A Lightweight Binarized Convolutional Block Attention Module: B-CBAM

Shaoqing Wu [1,*] and Hiroyuki Yamauchi [2,*]

[1] Intelligent Information System Engineering, Fukuoka Institute of Technology, Fukuoka, Japan
[2] Department of Computer Science and Engineering, Fukuoka Institute of Technology, Fukuoka, Japan
Email: bd24201@bene.fit.ac.jp (S.W.); yamauchi@fit.ac.jp (H.Y.)
*Corresponding author

***Abstract*—This study proposes a new channel and spatial attention mechanism module called the Binarized Convolutional Block Attention Module (B-CBAM), which is characterized by binarized channel attention feature map, optimized activation function for the binarization, and spatial attention feature map. Unlike the conventional Convolutional Block Attention Module (CBAM), the binarized attention feature map and the best choice of the activation function for the binarization allow accelerating the learning curve and eliminating the need for Max pooling and multilayer perceptron. The binarized convolution significantly reduces the computational load required by the attention module, with memory requirements being only 1/32 of those for full-precision weights. After training for 100 epochs, the average error rates improved by 21% on CIFAR-10 and 9% on STL-10. The optimized B-CBAM model, after parameter tuning, achieves error rates very close to those of the non-quantized model, with an error rate of around 9% on CIFAR-10.**

***Keywords*—binarized, residual model, attention mechanism, Convolutional Neural Network (CNN) efficiency, feature maps binarization**

I. Introduction

Convolutional Neural Networks (CNNs) [1–5] have revolutionized various computer vision applications and are now used as the primary architecture in most computer vision systems. Modern high-performance CNNs are often composed of repeated blocks with the same structure, leveraging the principles of residual learning [6] and utilizing depth wise separable convolutions [7]. These networks typically employ 32-bit floating-point representations for weights and activations. However, deploying these models on devices with strict power and memory footprint constraints poses significant challenges. Although CNN models are powerful tools, several challenges must be addressed for broader application. Nevertheless, existing methods rarely consider the interplay between binarization and attention modules, leaving a gap in simultaneously achieving strong performance and high efficiency in binarized CNNs.

Model Quantization is a technique that converts the high-precision weights and activation values in deep learning models to lower-precision representations. Typically, model quantization transforms 32-bit Floating-Point numbers (FP32) into lower bit-width representations such as 16-bit Floating-Point (FP16), 8-bit Integers (INT8), or even lower. This process significantly reduces the model's storage requirements and computational complexity without substantially degrading its performance. Below 8-bit, it is currently mainly used in academia, but it is also partially supported in the industry. When the precision decreases to 1-bit, this is called binarization, and computation can be performed using bitwise operations. XNOR-Net [8] is a classic 1-bit quantization network that quantizes the weights and activations in the network to –1 and 1. Binary computation allows matrix multiplication to be replaced with bitwise operations and addition. Introducing an optimized scaling factor, which compensates for the information loss caused by binarization, allows the model to retain a high level of expressiveness even after quantization.

The Attention Mechanism [9–11] that has attracted the most attention in recent years, is a technique in deep learning designed to mimic the human process of focusing attention, enabling models to automatically concentrate on important features when handling complex data. By assigning "attention" to critical features, the attention mechanism can enhance the performance and efficiency of models. The Squeeze-and-Excitation (SE) Module, introduced by Hu *et al.* [12], is a method that enhances the representational capacity of a network by adaptively recalibrating channel-wise feature responses [12–14]. The Convolutional Block Attention Module (CBAM), proposed by Woo *et al*. [15], is a module that combines channel attention and spatial attention to further enhance feature representation. The SE Module focuses solely on the channel dimension, enhancing feature representation by recalibrating the importance of each channel. In contrast, the CBAM Module extends this concept by incorporating spatial attention, allowing it to attend to both channel and spatial dimensions. Overall, the CBAM module offers a more detailed feature focus based on the SE module but comes with increased computational costs.

However, the conventional SE and CBAM module is designed with high-precision 32bit floating-point numbers

Manuscript received December 19, 2024; revised January 23, 2025; accepted March 12, 2025; published May 22, 2025.

and the attention feature maps are represented by the high-precision numbers. In addition, they use Sigmoid based activation function. We found that using the Sigmoid function in the channel attention module worsens the performance in the binarized operation. As a result, the conventional SE poses a critical challenge of slow learning curves [16, 17]. We also found that even if the conventional CBAM sacrifices extra additional operations (e.g., Parallel processing of Average Pooling and Max Pooling and Multilayer Perceptron), it will not be effective anymore for the binarized cases. This means that a different approach will be prerequisite for improving the performance over the conventional SE and CBAM for the binarized operations. Convolutional Block Attention Module for Binarized ResNet14-Wide based on previous experiments [16, 17].

To bridge this gap, we propose a novel lightweight Binarized Convolutional Block Attention Module (B-CBAM) that is specifically tailored to handle the constraints of 1-bit quantization without compromising model expressiveness. The novelty and main contributions of this paper are summarized as follows:

(1) We demonstrated for the first time the speed up effects on the learning curve by introducing the binarized feature map method to both of the channel and the spatial attention blocks. It is finally found that it is effective for the channel attention but counterproductive for the spatial attention.
(2) We also found that 1) the combination of Max and Average Pooling and associated multilayer perceptron doesn't work well and 2) the HardTanh activation function is the better choice than Sigmoid that conventionally being used in SE and CBAM.
(3) Based on new findings above in (1) and (2), we have proposed for the first time the new binarized attention module, referred to as B-CBAM, as shown in Fig. 1. Table I shows the comparison of SE, CBAM and B-CBAM attention mechanisms.
(4) To further optimize the deployment of ResNet14 on resource-limited devices, we applied binarization to the model. Binarization converts the model's weights and activation values from high-precision floating-point numbers (e.g., 32-bit) to lower-precision representations (e.g., 1-bit), significantly reducing the model's storage requirements and computational complexity. This not only lowers the model's storage footprint but also accelerates inference speed, allowing the model to run efficiently on devices with limited power and memory.
(5) We evaluated the effectiveness of B-CBAM on the CIFAR-10 and STL-10 datasets, and generated heat maps to verify the feasibility of the attention mechanism.

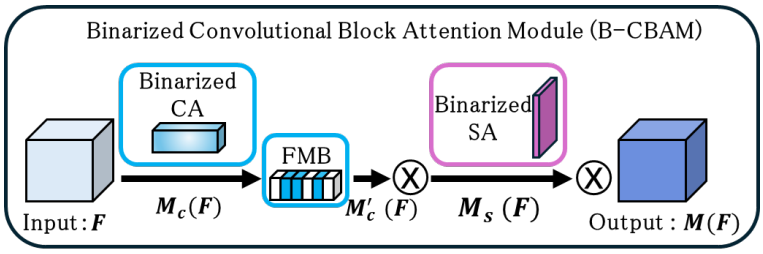

Fig. 1. Proposed B-CBAM: A binarized approach to convolutional block attention module.

TABLE I. COMPARISON OF SE, CBAM, AND B-CBAM ATTENTION MECHANISMS

Attention	Types	Floats	Pooling	Activation Function
Squeeze-and-Excitation (SE)	Channel	32bit	AvgPool	Sigmoid
Convolutional Block Attention Module (CBAM)	Channel and Spatial	32bit	AvgPool and MaxPool	Sigmoid
Binarized Convolutional Block Attention Module (B-CBAM)	Channel and Spatial	1bit	AvgPool	HardTanh and Sigmoid

The remainder of this article is organized as follows: In Section II, we briefly review the related work. In Section III, we present various optimization methods for CBAM, including the impact of activation functions and pooling operations on the accuracy curve, and demonstrate the superiority of our B-CBAM over CBAM. Section V discusses the implementation of enhanced attention mechanisms and Section IV provides a summary.

II. RELATED WORK

A. Binarized Quantization

The mainstream quantization methods currently include Post-Training Quantization (PTQ) [18], Dynamic Quantization, and Mixed Precision Quantization [19], among others. In this study, Quantization-Aware Training (QAT) [20] was used. The binarization quantization process is simulated during training, where the model uses low-precision representations (1-bit) during the forward pass, while still using floating-point numbers to update the weights during the backward pass. This approach allows the model to gradually adapt to the effects of quantization during training, thereby minimizing the accuracy loss caused by quantization.

Our binarization method constrains the model's weights to binary values of −1 and 1, just like BNNs [21]. This approach not only significantly decreases the model's storage requirements but also simplifies the computational process, as binary weights can leverage efficient bitwise operations for rapid calculations.

B. Miniaturization Model Optimization

ResNet (Residual Network) is a deep convolutional neural network architecture introduced by He *et al*. [6], designed to address the issues of vanishing gradients and model degradation that occur as network depth increases. ResNet achieves this by incorporating Residual Blocks, which include skip connections that allow information to flow directly from earlier layers to deeper layers. This structure effectively mitigates the challenges associated with training very deep networks, enabling the construction of deeper and more complex models.

Zagoruyko and Komodakis [22], the author of Wide ResNet (WRN), proposed that by reducing depth and increasing the number of channels (width) in each layer, the model can maintain or even improve performance while significantly reducing training time. They demonstrated that in certain scenarios, increasing network width is more effective than increasing depth, especially in reducing overfitting and enhancing the model's generalization ability. This model shares a similar design philosophy with the ResNet14-wide employed in this study, which improves expressiveness by widening the network rather than deepening it, making it equally important for deployment on resource-constrained devices.

Building upon the ResNet architecture, we designed a more compact ResNet14-wide model to cater to the needs of resource-constrained environments such as mobile devices and embedded systems. ResNet14-wide achieves model lightweighting by reducing the network depth and adjusting the number of channels in each layer, while striving to maintain the performance of the original ResNet. Specifically, ResNet14 removes the last residual block of ResNet18 and adjusts the input channels to 80, 160, and 320, significantly reducing the model's parameter count and computational complexity. In contrast, the original ResNet18 model has input channels of 64, 128, 256, and 512 respectively. Despite the reduced depth, we widened the network by increasing the number of channels to maintain its expressive power. This wider design ensures that ResNet14-wide can effectively operate in resource-limited devices without compromising accuracy, achieving a balance between model efficiency and performance.

To accommodate this binarization approach, we have replaced the original Rectified Linear Unit (ReLU) and Sigmoid activation functions in ResNet with the HardTanh activation function. HardTanh is a simplified version of the Tanh function, with its output range constrained between [−1, 1]. This modification offers several key advantages. Firstly, the output range of HardTanh aligns with the binary weight values, ensuring minimal error introduction during quantization, thereby enhancing model compatibility. Secondly, compared to ReLU and Sigmoid, HardTanh has lower computational complexity, making it especially suitable for hardware accelerators and improving computational efficiency. Additionally, HardTanh provides a linear region, avoiding the zero-output issue of ReLU in the negative range, thereby enhancing training stability. These improvements enable the ResNet14 model to significantly reduce computational and storage requirements while maintaining high accuracy, making it highly suitable for efficient deployment on resource-constrained AI devices.

C. Different Channel Attention Module in CBAM

Squeeze-and-Excitation (SE) modules and Convolutional Block Attention Module (CBAM) are attention mechanisms used to improve the performance of convolutional neural networks, but they differ in how they implement channel attention. The SE module performs global average pooling on the input features to obtain a global descriptor for each channel. It uses a bottleneck structure composed of two fully connected layers (reducing dimensions first and then increasing them) to learn the importance weights of each channel. The learned channel weights are multiplied channel-wise with the original features to recalibrate them. In the Eq. (1), the variable F is used to represent the input feature map, W_0 and W_1 are weight matrices of two fully connected layers used for channel weight recalibration, handling the dimensionality reduction and expansion of features.

$$M_c(F) = Sigmoid\big(AvgPool(F)\big) = Sigmoid\left(W_1\left(ReLU\left(W_0\left(F_{avg}^c\right)\right)\right)\right) \quad (1)$$

CBAM uses both global Average Pooling (AvgPool) and global Max Pooling (MaxPool) to generate two different channel descriptors. These two descriptors are passed through shared fully connected layers to obtain the attention weights for each channel. The channel weights are multiplied channel-wise with the original features to enhance the feature representation. After channel attention, CBAM further computes spatial attention through convolution operations to highlight important spatial locations. In the Eq. (2), W_0 and W_1 are shared for both inputs and ReLU activation function is followed by W_0.

$$M_c(F) = Sigmoid\left(MLP\big(AvgPool(F)\big) + MLP\big(MaxPool(F)\big)\right) = Sigmoid\left(W_1\left(W_0\left(F_{avg}^c\right)\right) + W_1\big(W_0(F_{max}^c)\big)\right) \quad (2)$$

In summary, the SE module focuses on channel-wise attention using global average pooling to learn how to recalibrate channel features. The CBAM module, on the other hand, utilizes both average and max pooling to gather richer feature information and introduces a spatial attention mechanism, allowing the model to focus on important features across both channel and spatial dimensions. Although both types of attention mechanisms can significantly improve model performance in conventional precision mode, the extra pooling operations may lead to instability in binarized and quantized models. Therefore, we will conduct experiments to verify which method can better adapt to the binarized ResNet. In our research, we found that channel attention modules like SE, which only use AvgPool, are sufficient to meet the needs of model miniaturization. We also noticed that using the

HardTanh function in the channel attention module performs better than using the Sigmoid function in this study. The channel attention calculation formula in this study, as shown in Eq. (3). And their differences are shown in Fig. 2.

$$M_c(F) = HardTanh\big(AvgPool(F)\big) = HardTanh\left(W_1\left(ReLU\left(W_0\left(F_{avg}^c\right)\right)\right)\right) \quad (3)$$

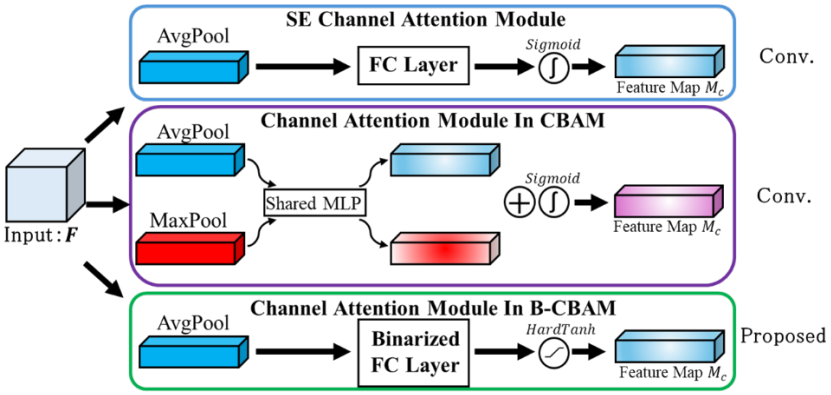

Fig. 2. The structural difference between SE attention and CAM attention in CBAM.

D. Spatial Attention Module for Binarized ResNet14-Wide

The spatial attention submodule in the CBAM module is used to capture important regions in the spatial dimension of the feature map, thereby helping the model focus better on key parts of the image. First, the input is a feature map from the previous layer with dimensions. To generate the spatial attention map, it is necessary to aggregate the channel dimension to extract information in the spatial dimension. Typically, average pooling and max pooling are used to extract global information from the feature map. The input feature map is pooled along the channel dimension using both average pooling and max pooling, resulting in two feature maps: F avg and F max. These two feature maps are then combined through a concatenation operation, forming a two-layer feature map F cat. Next, a binarized 3×3 convolution operation is applied to the concatenated feature map, along with an activation function, to generate the spatial attention map. Finally, the generated spatial attention map is multiplied element-wise with the input feature map, thereby highlighting important spatial regions in the feature map while downplaying less important ones. The spatial attention can be expressed by the Eq. (4):

$$M_s(F) = Sigmoid\left(f^{3\times3}([AvgPool(F)];[MaxPool(F)])\right) \quad (4)$$

where F is the input feature map, [;] denotes the concatenation operation, and f represents a binarized convolution operation. In this study, additional pooling operations introduce extra computational load. We demonstrated that using only AvgPool is sufficient to meet the needs of model miniaturization, as shown in Eq. (5).

$$M_s(F) = Sigmoid\left(f^{3\times3}[AvgPool(F)]\right) \quad (5)$$

E. Attention Optimization: Feature Map Binarization

To further optimize the attention mechanism, we also refined the channel attention mechanism. In this study, during Quantization Aware Training (QAT), the attention weights are binarized only during the backward pass, and the weights gradually become binarized as model training progresses. To accelerate this process, we enforced binary quantization on the weights after they were output by the activation function. We refer to this method as Feature Map Binarization (FMB) [15, 16].

III. Experiment and Results

This section compares the classification performance of models with different parameter sizes under both non-quantized and binarized conditions, as well as after applying spatial and channel attention, on the CIFAR-10 and STL-10 datasets. Both datasets are 10-class image classification tasks, with CIFAR-10 using 32×32 RGB images as input and STL-10 using 96×96 RGB images. For the STL-10 dataset, we only used the labeled portion for training. This comparison aims to validate our model's classification capabilities on low-resolution and high-resolution images.

Optimization algorithms play a crucial role in the training performance of neural networks. In BNNs, it is mentioned that using Adam as the optimizer for binarized models outperforms SGD. Additionally, our research has found that training our binarized models with the SGD optimizer results in convergence issues. Therefore, we employed Adam to train the binarized models in this study, while using SGD to train the standard FP32 models. This approach ensures stable and efficient training across different model architectures.

The Adam optimizer performs better in large-batch training by dynamically adjusting the learning rate. For the CIFAR-10 dataset, we applied dataset normalization and set a batch size of 256, which required approximately 6 GB of GPU memory during training. Similarly, for the STL-10 dataset, we used dataset normalization and a batch size of 128, resulting in approximately 16 GB of GPU memory usage during training. These configurations were chosen to balance training efficiency and resource utilization,

ensuring effective model convergence across different datasets.

Although this study focuses on the miniaturization of AI models, we employed a server equipped with an AMD R9 5950x 16-core processor, 64 GB of DDR5 memory, and an NVIDIA RTX 4090 GPU with 24 GB of memory to ensure smooth and efficient model training. This robust hardware configuration facilitated the training of our models without encountering computational bottlenecks, thereby allowing us to thoroughly evaluate the performance and scalability of the proposed miniaturized architectures on both CIFAR-10 and STL-10 datasets. Python version is 3.7.16 and PyTorch version is 1.10.1.

In this section, all the presented results are based on multiple repeated measurements under identical conditions. We randomly selected ten of these measurements and averaged them for comparison, thereby ensuring the reproducibility of the proposed method. We assess the effectiveness of our proposed method by observing improvements in the learning curve and enhancements in model accuracy.

A. Model Optimization: Balancing Parameter Count and Accuracy in ResNet Variants

Table II compares different ResNet models based on their parameter counts and Top-1 error rate on the CIFAR-10 and STL-10 datasets. To optimize our model for efficiency without significantly compromising accuracy, we aimed to reduce the complexity of ResNet18. After simplifying ResNet18, we developed ResNet14 and ResNet14-Wide models, which have fewer parameters but also exhibit some loss in accuracy. Additionally, we applied binarization techniques to further reduce computational costs. However, this binarization led to a slight decrease in accuracy due to the reduction in precision of the model weights and activations. Despite this, ResNet14-Wide with binarized convolutional layers still achieved a balance between parameter count, computational efficiency, and precision, making it a suitable choice for classification tasks on both CIFAR-10 and STL-10 datasets.

TABLE II. COMPARISON OF RESNET MODELS (FP32): PARAMETERS AND TOP-1 ERROR RATE ON CIFAR-10 AND STL-10 DATASETS

Models (FP32)	Param.	CIFAR-10 Top-1 Acc. (%)	STL-10 Top-1 Acc. (%)
ResNet14	2.78M	6.92	18.35
ResNe14-Wide	4.34M	6.33	16.32
ResNet18	11.18M	6.12	16.13

B. Channel Attention Mechanisms for Binarized ResNet14-Wide

Channel Attention is a mechanism used in deep learning models to enhance feature representation, especially in Convolutional Neural Networks. It weights the channels of feature maps to highlight important feature channels, thereby improving the model's performance.

Global average pooling or max pooling is performed over the spatial dimensions of the input feature map to obtain a global description for each channel. This step captures the overall information of each channel. The aggregated global information is passed through a series of fully connected layers (which may include nonlinear activation functions and normalization operations) to generate attention weights for each channel. These weights reflect the importance of each channel. Then generated channel weights are multiplied with the original feature map on a channel-by-channel basis to obtain a re-weighted feature map, thereby emphasizing important channels and suppressing less important ones.

The SE Attention Mechanism is a type of channel attention. This mechanism consists of two main steps: Squeeze and Excitation. First, in the squeeze step, global average pooling is applied to aggregate the spatial information of each channel into a single value, generating a channel descriptor. Next, in the excitation step, two fully connected layers and activation functions are used to learn the importance weights of each channel. Finally, these weights are used to readjust each channel of the original feature map, thereby enhancing key features and suppressing unimportant ones. The SE Attention Mechanism can significantly improve the model's representational capacity and classification performance while adding only minimal computational overhead.

The Channel Attention Module (CAM) in CBAM (Convolutional Block Attention Module) generates more comprehensive channel weights by combining global average pooling and global max pooling. Specifically, CBAM first applies AvgPool and MaxPool separately to the input feature map to obtain two distinct global descriptors. These descriptors are then processed through a shared Multi-Layer Perceptron (MLP), and their outputs are summed and passed through a Sigmoid activation function to produce the final channel attention weights. This approach allows CBAM to capture both the average features and the most salient features of each channel, providing more holistic information compared to SE attention, which uses only GAP. Fig. 3 shows the SE module adapted for our binarized network, using the HardTanh function as the activation function to quantize the channel attention weights to −1 and 1. This approach is more suitable for binarized networks compared to the original SE module, which uses the Sigmoid function. Similarly, Fig. 4 demonstrates the changes in the CAM module after using the HardTanh activation function.

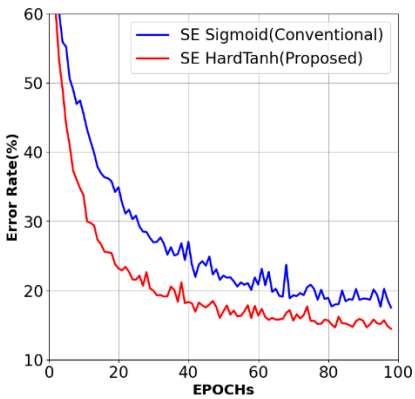

Fig. 3. The performance of the SE module using different activation functions.

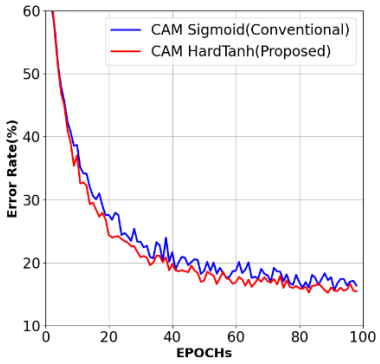

Fig. 4. The performance of the CA module using different activation functions.

We also found that using both MaxPool and AvgPool in the CAM module, compared to the SE module's sole use of AvgPool, resulted in slower model convergence. This indicates that in small-scale binarized networks, more pooling operations may not necessarily improve model performance. On the contrary, it could increase the model's complexity, making a previously simple network harder to converge. As shown in the Figs. 5 and 6, our FMB binarization method significantly improves the performance of both attention mechanism modules. However, we noticed that the SE module with the FMB method still slightly outperforms the CAM module.

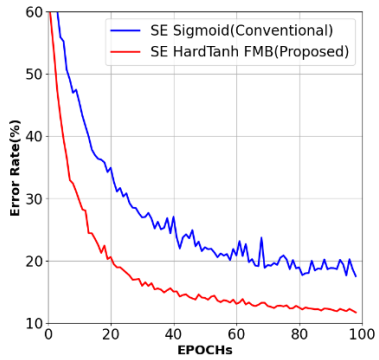

Fig. 5. The performance of the SE module using FMB method.

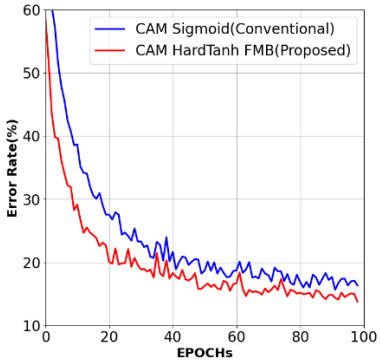

Fig. 6. The performance of the CA module using FMB method.

C. *Spatial Attention Module for Binarized ResNet14-Wide*

For the selection of activation functions and pooling operations, we experimented with HardTanh and Sigmoid functions for activation, and tested AvgPool and MaxPool individually as well. The results are shown in Figs. 7 and 8. We found that in this study, the combination of AvgPool and Sigmoid worked best for spatial attention. Since the model weights are quantized to −1 and 1, if the output attention feature map also consists of −1 and 1, multiplying it with the original input could cause a sign reversal issue. This would result in areas that the attention mechanism originally ignored (with a weight of −1) becoming incorrectly emphasized due to the multiplication of −1 by −1, resulting in 1. As for the pooling operation, because the model itself has a very small number of parameters, and these parameters are binarized, excessive pooling operations may not bring a significant improvement in accuracy.

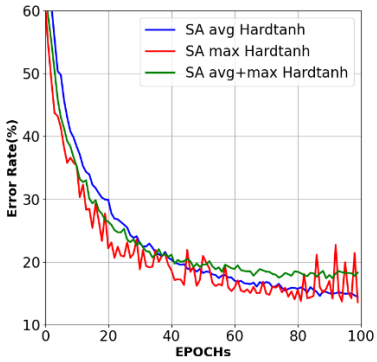

Fig. 7. The performance of the SA module using different activation functions and pooling.

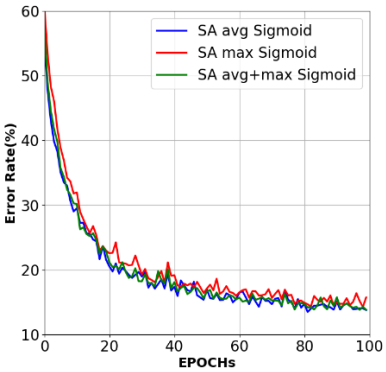

Fig. 8. The performance of the SA module using different activation functions and pooling.

We also observed that using the FMB method in spatial attention did not improve the model's performance. Since the model already has a shallow network depth, forced binarization does not enhance the attention mechanism's capability; rather, it damages the model's original performance. Therefore, we do not recommend using the FMB method in spatial attention. We are the first to demonstrate that introducing the binarized feature map method into both channel and spatial attention blocks accelerates the learning process. Our results indicate that while this method enhances the performance of the channel attention block, it proves counterproductive for the spatial attention block.

D. Convolutional Block Attention Module for Binarized ResNet14-Wide

Based on previous experiments, it can be concluded that using only AvgPool in channel and spatial attention, compared to the original CBAM's two pooling operations, can reduce the computational load by half, helping to lower the FLOPS of the attention mechanism. For the optimized B-CBAM (Binarized-Convolutional Block Attention Module) in this study, the HardTanh function is used as the activation output in the channel attention, and the FMB method is employed to enhance the channel attention. For spatial attention, the two pooling operations from the original CBAM are simplified to only AvgPool.

As shown in Table III, the best accuracy achieved after an average of 100 epochs increased by 21% on CIFAR-10 and 9% on STL-10, respectively. We define the mean difference in Top-1 error rate as Δ = Err. $_{\text{CBAM}}$ − Err. $_{\text{B-CBAM}}$. A positive Δ thus indicates that B-CBAM outperforms CBAM. On the CIFAR-10 dataset, a Welch's t-test revealed a highly significant mean difference in Top-1 error rates Δ between CBAM and B-CBAM, with t = 15.48, $p = 1.94\times10^{-10}$, and a 95% confidence interval of (2.41,3.18). Since this interval lies entirely above zero, our results indicate that B-CBAM outperforms CBAM on CIFAR-10. A similar trend was observed for the STL-10 dataset, where the t-test yielded t = 6.88, $p = 1.49\times10^{-5}$, and a 95% confidence interval of (2.35, 4.53). Again, the interval does not overlap zero, demonstrating a statistically significant difference in favor of B-CBAM.

TABLE III. THE COMPARISON OF THE ERROR RATE ACHIEVED BY RESNET14-WIDE WITH B-CBAM AFTER 100 EPOCHS

Models(1bit)	CIFAR-10 Top-1 Error (%)	STL-10 Top-1 Error (%)
B-CBAM ResNet14-Wide	12.21	35.32
CBAM ResNe14-Wide	15.53	38.84

Figs. 9 and 10 illustrate the performance improvement of our B-CBAM compared to the original CBAM on CIFAR-10 and STL-10. Compared to the 32-bit CBAM model, our B-CBAM reduces memory usage by 1/32. Additionally, by using only AvgPool instead of two pooling operations, it also reduces the computational load by half.

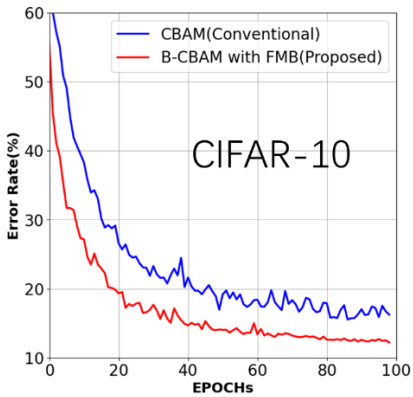

Fig. 9. The learning curve of the model on CIFAR-10.

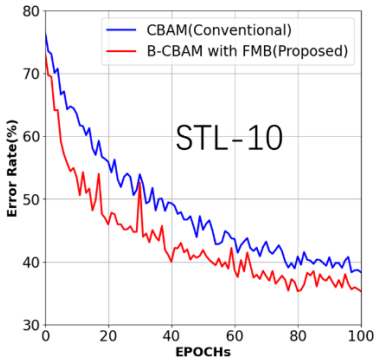

Fig. 10. The learning curve of the model on STL-10.

IV. DISSCUSION

Figs. 11 and 12 uses EigenCAM [23] to generate attention heatmaps to show the differences between our B-CBAM model and the original CBAM model. The deeper the red color, the more the model focuses on that position; conversely, the deeper the blue color, the less it focuses on that position. It can be seen that both attention mechanisms change the way the model processes information. In the images of the airplane and truck, after adding the CBAM module, the model pays more attention to the contours or background of the objects rather than the target objects themselves.

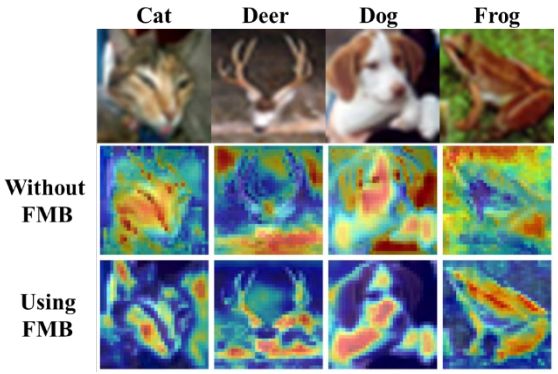

Fig. 11. EigenCAM attention heatmaps comparing the model with using FMB method.

In Fig. 11, after applying the channel attention FMB method in our B-CBAM, as seen in the heatmaps of the cat, deer, dog, and frog, it is evident that the red focus areas are deeper in color and fewer in number. This indicates that the model's focus areas are more concentrated, precise, and efficient. The blue areas are also deeper compared to those in the CBAM model, which shows that the regions irrelevant to the model are suppressed. This is precisely the function of the attention mechanism, and we have optimized and enhanced this effect.

The FMB method approach sets a single threshold (currently 0, since our binarized model maps values to −1 or 1) to forcibly binarize the feature map. It might be possible to introduce multiple thresholds to achieve a piecewise binarization of the feature map, which could serve as a potential solution.

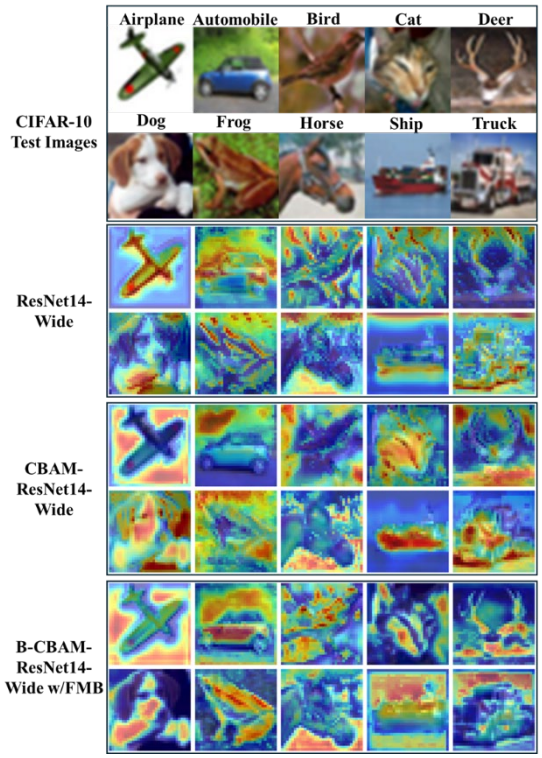

Fig. 12. EigenCAM attention heatmaps comparing CBAM and B-CBAM models.

V. Conclusion

This study proposes a Binarized-Convolutional Block Attention Module (B-CBAM), which includes channel and spatial attention mechanisms and can be used in binarized convolutional neural networks to accelerate training and improve accuracy. Compared to the original Convolutional Block Attention Module (CBAM), B-CBAM significantly reduces computational cost by using binary convolutions and by adopting only global average pooling instead of both global max pooling and global average pooling. Specifically, the binary convolutions greatly reduce the computational load of convolution operations, and the simplified pooling strategy further decreases the computational overhead of pooling operations. Considering the computational load of each component, our B-CBAM module achieves a more efficient attention mechanism than the original CBAM.

The thoroughly optimized B-CBAM-ResNet14-Wide model can even achieve performance close to the non-quantized (32-bit) model under 1-bit quantization (approximately 9% Top-1 error rate on the CIFAR-10 dataset). After training for 100 epochs, the average best accuracy attainable is improved by 21% (CIFAR-10) and 9% (STL-10) compared to the original CBAM's ResNet14-Wide. This study's primary limitation is that it was validated using only two datasets, which may restrict the generalizability of the findings. In future research, we plan to examine the method's effectiveness on a broader range of datasets with diverse characteristics, ensuring a more comprehensive evaluation of its performance across various domains.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

S. Wu: Conceived and designed the B-CBAM approach, performed the experiments, conducted data analysis, and prepared the initial manuscript draft.

H. Yamauchi: Provided conceptual guidance, supervised the overall research process, and reviewed and edited the manuscript.

All authors have read and approved the final manuscript version.

Acknowledgment

We are particularly grateful to the members of Professor Hiroyuki Yamauchi's lab for their invaluable assistance and support throughout this research. Additionally, we extend our gratitude to Fukuoka Institute of Technology for providing the necessary resources and infrastructure that facilitated the successful completion of this study.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 60, no. 6, pp. 84–90, Jun. 2017.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, arXiv:1409.1556, 2014.

[5] W. Samek, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," arXiv preprint, arXiv:1708.08296, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[7] D. Chenna, "Evolution of Convolutional Neural Network (CNN): Compute vs memory bandwidth for edge AI," arXiv preprint, arXiv:2311.12816, 2023.

[8] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 525–542.

[9] A. Vaswani, N. Shazeer, N. Parmar *et al*., "Attention is all you need," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[10] R. R. Saritha and V. Sangeetha, "Deep alternate kernel fused self-attention model-based lung nodule classification," *Journal of Advances in Information Technology*, vol. 15, no. 11, pp. 1242–1251, 2024.

[11] H. Pooja and M. P. P. Jagadeesh, "Integrated deep learning with attention layer based approach for precise biomedical named entity recognition," *Journal of Advances in Information Technology*, vol. 15, no. 6, pp. 704–713, 2024.

[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7132–7141.

[13] M. Bayyapureddi, L. S. Harshitha, and A. T, "Enhancing bird species classification with PaliGemma and custom EfficientNet architectures," in *Proc. 2024 8th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Coimbatore, India, 2024, pp. 642–649.

[14] Y. Chen, W. Du, X. Duan, Y. Ma, and H. Zhang, "Squeeze-and-excitation convolutional neural network for classification of malignant and benign lung nodules," *Journal of Advances in Information Technology*, vol. 12, no. 2, 2021.
[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
[16] S. Wu and Y. Hiroyuki, "A speed-up channel attention technique for accelerating the learning curve of a binarized Squeeze-and-Excitation (SE) based ResNet model," *Journal of Advances in Information Technology*, vol. 15, no. 5, 2024.
[17] S. Wu and Y. Hiroyuki, "A binarized feature mapping technique for enhancing Squeeze-and-Excitation (SE) channel attention mechanism," *International Journal of Machine Learning*, vol. 15, no. 1, pp. 23–28, 2025.
[18] B. Jacob, S. Kligys, and B. Chen, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 2704–2713.
[19] P. Micikevicius, S. Narang, J. Alben, "Mixed precision training," arXiv preprint, arXiv:1710.03740, 2017.
[20] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," arXiv preprint, arXiv:1806.08342, 2018.
[21] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1," arXiv preprint, arXiv:1602.02830, 2016.
[22] S. Zagoruyko and N. Komodakis, "Wide residual networks," arXiv preprint, arXiv:1605.07146, 2016.
[23] M. B. Muhammad and M. Yeasin, "Eigen-CAM: Class activation map using principal components," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2020, pp. 1–7.