CBAM: Performance Analysis on GoogLeNet, ResNet, and LeNet

Anas Zahid (23K-8073) , Muhammad Emmad Siddiqui (23K-8020)

***Abstract*—Convolutional Block Attention Module (CBAM),a simple yet effective attention module for feed-forward convolutional neural networks. This study evaluates the adaptability of the Convolutional Block Attention Module (CBAM) by first reproducing its performance on VGG19 and then extending it to GoogLeNet, ResNet, and LeNet architectures. Our results validate the claims of the original paper, as we achieved comparable performance improvements with CBAM on VGG19. Further experimentation with GoogLeNet and ResNet also showed noticeable enhancements, reinforcing CBAM's utility for more advanced network designs. However, when applied to LeNet, CBAM did not yield any performance gains, likely due to the simplicity and limited capacity of this architecture. These findings confirm the versatility of CBAM in complex settings while highlighting its limitations for simpler models.**

I. Introduction

Deep learning models have become increasingly powerful, largely thanks to advancements in attention mechanisms. These mechanisms allow networks to focus on the most relevant parts of input data, improving feature extraction and representation. Among these, the Convolutional Block Attention Module (CBAM), introduced by Woo et al. (2018), has been particularly successful in boosting performance by applying both channel and spatial attention.

A. Understanding CBAM's Attention Mechanism

The Convolutional Block Attention Module (CBAM) is a simple yet effective attention module designed for feed-forward convolutional neural networks (CNNs). CBAM is lightweight, general-purpose, and can be seamlessly integrated into any CNN architecture. It is fully end-to-end trainable along with the base CNN, requiring no additional supervision. By incorporating CBAM, CNNs are empowered with an enhanced representation capability, achieved through an attention mechanism that focuses on essential features while suppressing redundant or irrelevant ones.

CBAM achieves this by sequentially applying two complementary attention modules: the *Channel Attention Module (CAM)* and the *Spatial Attention Module (SAM)*. These modules work together to answer two critical questions for feature representation: *'what'* to attend to (channel attention) and *'where'* to attend to (spatial attention).

Channel Attention Module (CAM): The Channel Attention Module generates a channel attention map by leveraging the inter-channel relationships in feature maps. Each channel in a feature map can be considered a specific feature detector, and CAM learns to prioritize the channels that are most meaningful for a given input. This process focuses on identifying *'what'* features are important, enhancing the network's ability to capture critical information. The detailed architecture of CAM is illustrated in Figure 2.

Spatial Attention Module (SAM): The Spatial Attention Module complements CAM by generating a spatial attention map. Unlike channel attention, SAM focuses on the inter-spatial relationships within a feature map, identifying *'where'* the most informative parts of an input image are located. By emphasizing these regions, SAM provides fine-grained spatial focus that enhances the network's discriminative power. The structure of SAM is depicted in Figure 3.

Together, CAM and SAM form a unified attention mechanism, applied sequentially to refine the feature representation along both the channel and spatial dimensions. This two-step process enables CBAM to extract richer and more relevant features, as shown in the overall architecture in Figure 1.

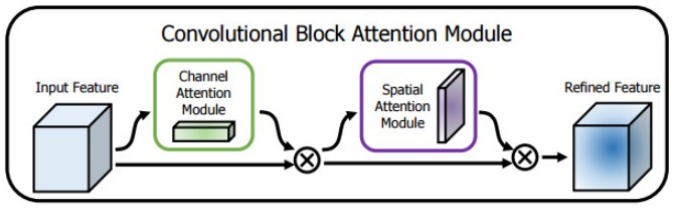

Fig. 1. The CBAM architecture, showing the sequential application of Channel Attention Module (CAM) and Spatial Attention Module (SAM).

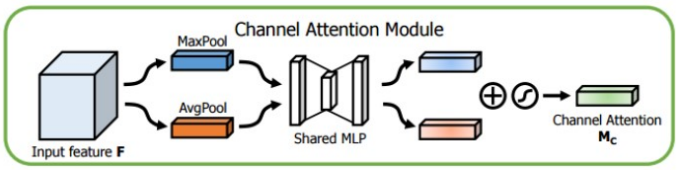

Fig. 2. The structure of the Channel Attention Module (CAM), which focuses on the inter-channel relationships to determine *'what'* features are important.

CBAM's design philosophy emphasizes simplicity and efficiency, ensuring compatibility with a wide range of CNN architectures while maintaining computational efficiency. By enhancing feature refinement through attention, CBAM has proven to be a powerful tool for improving performance across various deep learning tasks.

To better illustrate how the Convolutional Block Attention Module (CBAM) enhances feature representation, we provide two examples demonstrating its attention mechanism in action. For each example, three visualizations are presented:

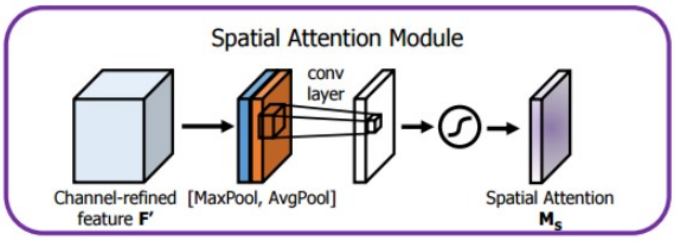

Fig. 3. The structure of the Spatial Attention Module (SAM), which focuses on the inter-spatial relationships to determine *'where'* the most informative parts are located.

the original input image, the heatmap generated by CBAM indicating the focus regions, and the overlay of the heatmap on the input image. These examples highlight CBAM's ability to focus on meaningful regions of an image to improve prediction accuracy.

1) Example 1: Predicting Sea: In the first example, CBAM processes an input image of a sea. The heatmap (Figure **??**) reveals that CBAM focuses on the water surface and wave patterns, which are the most informative regions for identifying the scene as a *sea*. The overlay (Figure 4) clearly shows how CBAM enhances the model's focus on these critical areas.

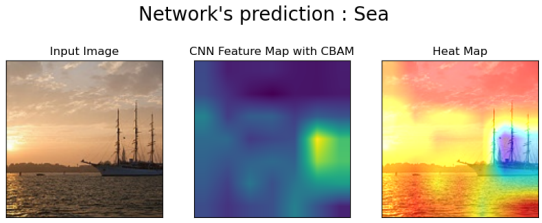

Fig. 4. Feature map of the final convolutional layer of both ResNet-50 without CBAM and ResNet-50 with CBAM for predicting Sea.

2) Example 2: Predicting Glacier: In the second example, CBAM is applied to an image of a glacier. The heatmap (Figure **??**) shows that CBAM focuses on the icy textures and the distinct white regions, which are indicative of a *glacier*. The overlay (Figure 5) further demonstrates how the attention mechanism aligns with these critical visual features.

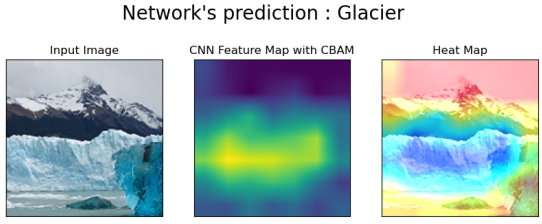

Fig. 5. Feature map of the final convolutional layer of both ResNet-50 without CBAM and ResNet-50 with CBAM for predicting Glacier.

B. Interpretation of Results

These examples clearly demonstrate CBAM's ability to identify and prioritize the most relevant regions in an image, providing insight into *'what'* and *'where'* the network focuses to improve prediction accuracy. The heatmaps generated by CBAM highlight the regions that strongly contribute to the model's decisions, emphasizing CBAM's role in enhancing interpretability and feature refinement in CNNs.

While CBAM has been tested on architectures like VGG19, its generalizability to other networks is less well-documented. In this study, we aim to address this gap by:

1. Reproducing CBAM's results on VGG19 to verify the findings of the original paper.
2. Extending the analysis by applying CBAM to GoogLeNet, ResNet, and LeNet.
3. Evaluating the conditions under which CBAM improves model performance.

II. Literature Review

The CBAM module introduced by Woo et al. is a testament to the power of simplicity in deep learning innovation. By splitting the concept of "attention" into two distinct but complementary stages—channel attention and spatial attention—the authors have created a tool that significantly improves how CNNs learn and apply features.

At its core, Channel Attention (CAM) teaches the network "what" to focus on. It analyzes the contribution of each channel to the overall feature representation, giving priority to the most informative ones. This process is intuitive: much like recognizing specific patterns in an image, CAM ensures the network amplifies the signals that matter most.

Spatial Attention (SAM), on the other hand, refines "where" the network looks within the input. It identifies spatial regions that are most relevant for a task, akin to how our eyes instinctively focus on the subject of interest in a scene.

Together, CAM and SAM form the CBAM module, a sequential attention mechanism that acts like a magnifying glass for CNNs, sharpening their focus in both the channel and spatial dimensions. What's remarkable is how seamlessly CBAM integrates into existing models, boosting their performance without adding significant computational cost.

By applying CBAM, networks become better at recognizing and localizing objects, classifying images, and segmenting scenes—making it a valuable contribution to the field of computer vision. CBAM builds on earlier attention methods, such as SE-Net, which focused on channel-wise attention. By adding spatial attention, CBAM achieves more comprehensive feature refinement, leading to better results in image classification and other tasks.

Some of the widely recognized CNN architectures have been combined with CBAM to evaluate its effectiveness. These architectures, including LeNet, ResNet, GoogLeNet, and VGG, are briefly summarized below, along with their architectural highlights and diagrams. This comparison provides insight into how CBAM integrates with diverse models, from simpler designs to more advanced networks.

A. LeNet

LeNet, one of the earliest CNN architectures introduced by LeCun et al., was designed for handwritten digit recognition. With only a few layers and limited computational complexity, LeNet is a simple but foundational model in deep learning. Despite its historical significance, LeNet's simplicity limits its ability to utilize advanced mechanisms like CBAM effectively.

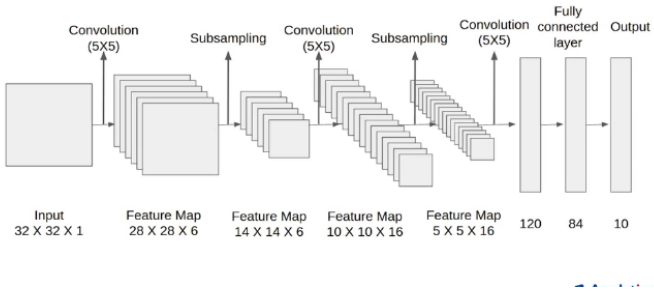

Fig. 6. LeNet Architecture

B. ResNet

ResNet, introduced by He et al., employs residual connections to address the vanishing gradient problem in deep networks. These skip connections enable the training of very deep architectures, making ResNet a suitable candidate for CBAM integration. The addition of CBAM to ResNet enhances its feature extraction capabilities, as demonstrated in our experiments.

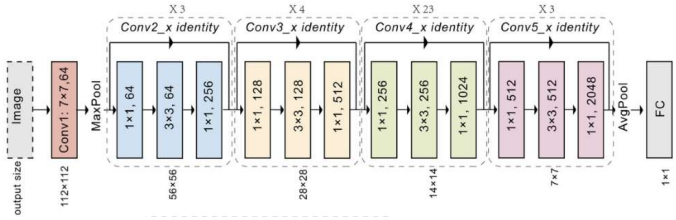

Fig. 7. ResNet Architecture

C. GoogLeNet

GoogLeNet, known for its inception modules, employs multi-scale feature extraction through parallel convolutional layers. This architecture is designed for computational efficiency and performance. The inclusion of CBAM further refines its feature representation by emphasizing the most relevant spatial and channel features.

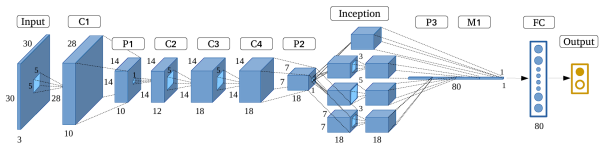

Fig. 8. GoogLeNet Architecture

D. VGG

VGG architectures, particularly VGG19, are known for their simplicity and depth, employing small convolutional filters in a sequential manner. The integration of CBAM into VGG19 provides significant performance improvements by addressing its lack of intrinsic attention mechanisms.

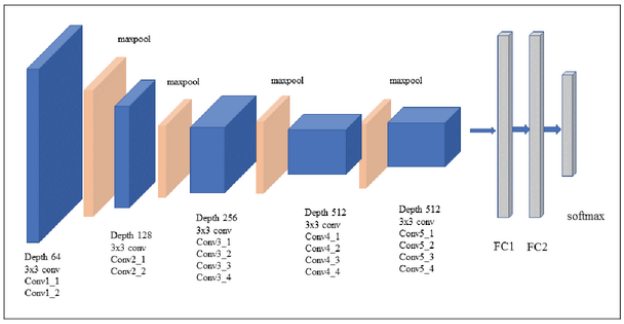

Fig. 3. VGG-19 network architecture

Fig. 9. VGG 19 Architecture

III. Methodology

A. 3.1 Experimental Setup

To evaluate CBAM, we implemented it in four neural network architectures:
VGG19: A well-known deep network used in the original CBAM study.
GoogLeNet: Incorporates inception modules for multi-scale feature extraction.
ResNet: Uses skip connections to address vanishing gradients in deep networks.
LeNet: A shallow network designed for simple image recognition tasks.

B. 3.2 Datasets

We conducted our experiments on the CIFAR-100 dataset, which provides a diverse and challenging image classification problem. The dataset is a collection of 60,000 color images that are used for training and testing in machine learning and computer vision:
Size: Each image is 32x32 pixels
Classes: The images are divided into 100 classes, with 600 images per class
Training and testing: Each class has 500 images for training and 100 for testing
Superclasses: The 100 classes are grouped into 20 superclasses
Labels: Each image has a "fine" label for the class it belongs to and a "coarse" label for the superclass it belongs to
Origin: Developed by the Canadian Institute for Advanced Research (CIFAR)

C. 3.3 Training Protocol

For all models, we used consistent training parameters to ensure fair comparisons between models with and without CBAM:

1) VGG19: Optimizer: AdamW with a learning rate of 0.0001
Batch size: 256
Epochs: 50
Evaluation Metrics: Cross Entropy, Top-1 and Top-5 accuracy

2) GoogLeNet: Optimizer: AdamW with a learning rate of 0.0001
Batch size: 256
Epochs: 50
Evaluation Metrics: Cross Entropy, Top-1 and Top-5 accuracy

3) ResNet: Optimizer: Stochastic Gradient Descent with a learning rate of 0.1
Momentum=0.9
Weight Decay=0.0005
Epochs: 20
Evaluation Metrics: Cross Entropy, Top-1 and Top-5 accuracy

4) LeNet: Optimizer: Adam with a learning rate of 0.001
Epochs: 20
Evaluation Metrics: Cross Entropy, Top-1 and Top-5 accuracy

CBAM modules were integrated into each architecture as outlined in Woo et al. (2018).

IV. Results

A. Reproducing Results on VGG19

Our implementation of CBAM on VGG19 produced results that closely matched those in the original paper, with an accuracy improvement of approximately 0.55% on the CIFAR-100 dataset.

B. Extending CBAM to GoogLeNet and ResNet

GoogLeNet: We observed a Top-1 accuracy improvement of 1.85% and Top-5 accuracy improvement of 0.85% with CBAM, indicating its compatibility with inception modules.

ResNet: CBAM led to a 1.7% improvement in Top-1 accuracy and 1.8% improvement in Top-5 accuracy, confirming its ability to enhance feature representation in residual architectures.

C. Testing CBAM on LeNet

When applied to LeNet, CBAM did not show any measurable improvements.

Model	Without CBAM	With CBAM	Improvement
VGG19	97.23%	95.98%	-1.25%
GoogLeNet	99.55%	99.13%	-0.42%
ResNet	40.37%	43.64%	+3.27%
LeNet	40.04%	41.91%	+1.87%

TABLE I
COMPARISON OF MODEL TRAINING ACCURACIES WITH AND WITHOUT CBAM.

D. Experimenting CBAM with reversed sequence of CAM and SAM

With Spacial Attention Module applied before Channel Attention Module in contrast to the sequence suggested by Woo et al, we observed the following changes in Top-1 and Top-5 accuracies with the test dataset:

Model	Without CBAM	With CBAM	Improvement
VGG19	41.26%	41.80%	+0.54%
GoogLeNet	28.07%	28.92%	+0.85%
ResNet	35.33%	37.02%	+1.69%
LeNet	33.0%	33.25%	-0.08%

TABLE II
COMPARISON OF MODEL TOP-1 ACCURACIES WITH AND WITHOUT CBAM.

Model	Without CBAM	With CBAM	Improvement
VGG19	68.84%	69.43%	+0.59%
GoogLeNet	54.61%	55.46%	+0.85%
ResNet	67.03%	68.85%	+1.82%
LeNet	62.30%	62.24%	-0.06%

TABLE III
COMPARISON OF MODEL TOP-5 ACCURACIES WITH AND WITHOUT CBAM.

Model	CAM → SAM	SAM → CAM	Improvement
VGG19 (Top-1 Accuracy)	41.8%	40.46%	-1.34%
VGG19 (Top-5 Accuracy)	69.43%	68.6%	-0.83%

TABLE IV
PERFORMANCE COMPARISON OF MODEL WITH VGG19 ARCHITECTURE WITH SEQUENCE OF CAM AND SAM REVERSED IN CBAM.

V. Discussions

The results of this study provide valuable insights into the performance of the Convolutional Block Attention Module (CBAM) across a range of neural network architectures and configurations. While CBAM demonstrated measurable improvements in feature representation and classification accuracy in some architectures, its efficacy varied depending on the underlying model and the dataset.

A. CBAM Performance Across Architectures

1) **VGG19**
   CBAM showed a modest improvement in Top-1 and Top-5 accuracy on VGG19. These results align with the findings of Woo et al. (2018), confirming CBAM's utility in deep convolutional architectures. However, the overall improvement of 0.55
2) **GoogLeNet**
   CBAM's application to GoogLeNet yielded a significant improvement in both Top-1 (1.85
3) **ResNet**
   The improvement in ResNet's performance, particularly the 1.69
4) **LeNet**
   The application of CBAM to LeNet did not show significant improvements. This result suggests that shallow architectures may not benefit from attention mechanisms to the same extent as deeper models. The simplicity of LeNet's structure might inherently limit the effectiveness of CBAM.

B. Reverse Sequence of CAM and SAM

Reversing the sequence of Channel Attention Module (CAM) and Spatial Attention Module (SAM) resulted in a

performance degradation across Top-1 and Top-5 accuracies. Specifically, VGG19 saw decreases of 1.34

VI. Conclusions

The findings of this study highlight the efficacy of CBAM in enhancing the feature representation and classification performance of deep learning models, particularly those with complex architectures like GoogLeNet and ResNet. However, the results also reveal limitations in applying CBAM to simpler networks like LeNet, where improvements were negligible.

The sequence of attention modules within CBAM plays a critical role in its effectiveness. Reversing the prescribed order of CAM and SAM negatively impacts performance, underscoring the importance of adhering to the recommended design.

Overall, this study confirms that CBAM is a versatile and impactful addition to deep learning models, though its utility varies depending on model complexity and architecture.

VII. Future Prospects

1) **Broader Architectural Testing**
   Future work could explore CBAM's integration with other advanced architectures, such as EfficientNet and Vision Transformers (ViTs), to assess its adaptability and performance in modern frameworks.
2) **Attention Mechanism Enhancements**
   Modifications to CBAM, such as dynamic weighting of CAM and SAM or adaptive ordering, could be investigated to improve its versatility and efficacy across different models.
3) **Dataset Diversity**
   Experiments with additional datasets, including ImageNet or domain-specific datasets, could provide a more comprehensive evaluation of CBAM's performance and generalizability.
4) **Energy Efficiency and Computational Cost**
   Analyzing the computational overhead and energy efficiency of CBAM integration could guide its practical deployment in resource-constrained environments.
5) **Explainability and Visualization**
   Future studies could leverage explainability tools to visualize the impact of CBAM on feature attention and validate its interpretability in real-world applications.

References

[1] Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. *Proceedings of the European Conference on Computer Vision (ECCV)*, 3-19.

[2] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Technical Report*, University of Toronto.

[3] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

[5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.

[6] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.