CS7015: Deep Learning — Project Proposal

CBAM: Convolutional Block Attention Module

Anoubhav Agarwaal (BE16B002) Ricardo Lucas (ME19F011)

1 Objective

The objective of this project is to extend the CBAM: Convolutional Block Attention Module [1] paper. We contrasted the performance of three separate variants of ResNet-50 [2]. Namely, vanilla (baseline), CBAM module, and the proposed module inspired by ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks [3]. We experiment on two datasets, namely, CIFAR-100 and CIFAR-10.

2 Introduction

The Convolutional Block Attention Module (CBAM) is an attention module for feed-forward convolutional neural networks. Given an intermediate feature map, CBAM sequentially infers attention maps along two separate dimensions, channel and spatial, so that each of these branches can learn 'what' and 'where' to attend. Then the attention maps are multiplied to the input feature map for adaptive feature refinement.

To compute the channel attention, average pooling and max-pooling are used to squeeze the spatial dimension of the input feature map and gathering the important information. To compute the spatial attention, we first apply the same pooling operations as before along the channel axis and concatenate them to generate an efficient feature descriptor. Applying pooling operations along the channel axis is shown to be effective in highlighting informative regions. Experiments from the authors show consistent improvements in classification and detection performances with various models, demonstrating the wide applicability of CBAM.

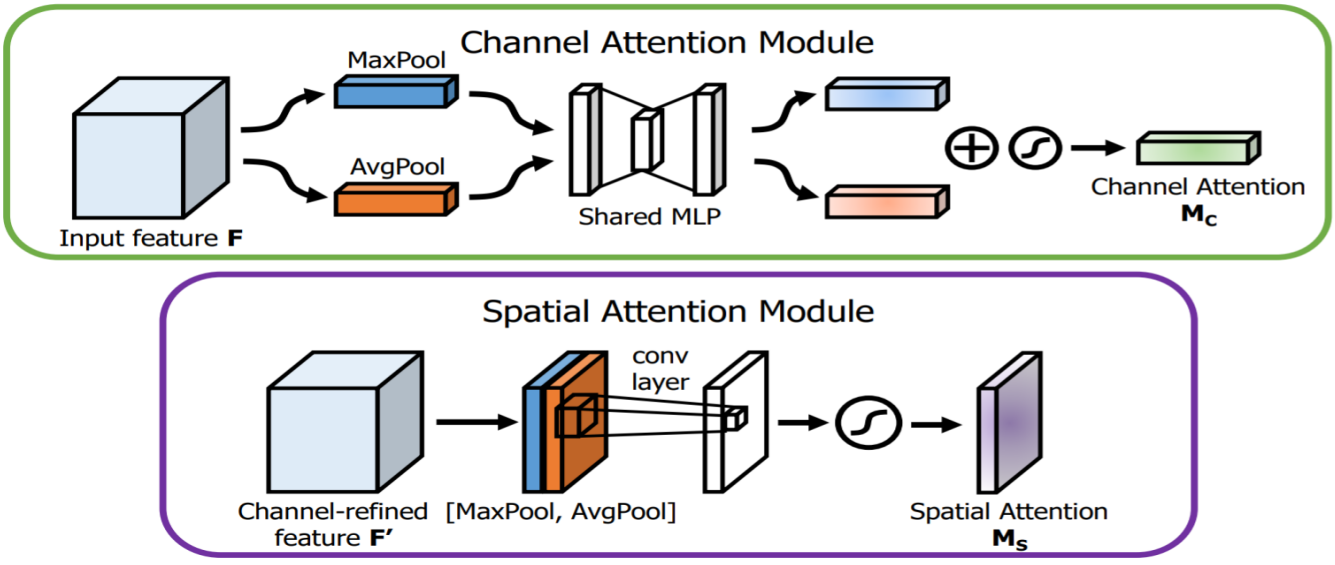

Figure 1: Diagram of each attention sub-module in CBAM. [1]

For our proposed model we borough an idea from the ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks paper [3]. In this paper the authors propose the Efficient Channel Attention (ECA) module by revisiting the channel attention module in SENet and replacing the multi-layer perceptron (MLP) by a fast 1D convolution with an adaptive kernel size (k) via a function of channel dimension. This module not only increases the performance relative to SENet and CBAM, but also does so with only adding k number of parameters.

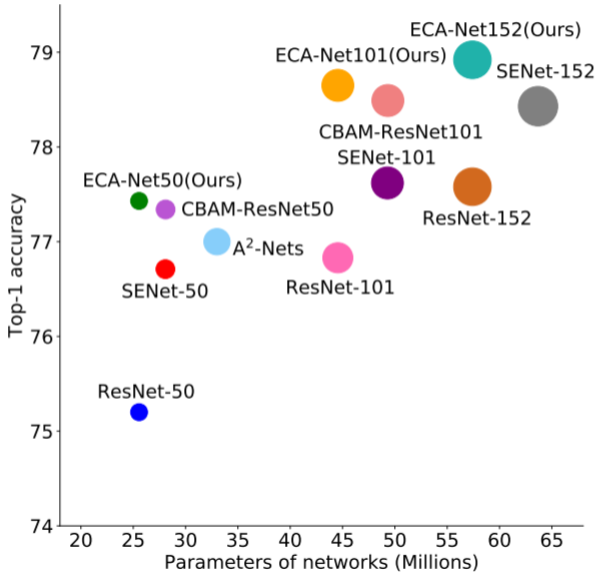

Figure 2: Comparison of various attention modules. [3]

In the end, our proposed module consisted of modifying the CBAM module by adding **two more types of pooling operations** and **replacing the shared MLP with an adapted fast shared 1D convolution from the ECA module**. The rationale behind picking two more pool types was similar to how the authors of CBAM incorporated max-pool in addition to average-pool, as seen earlier in Squeeze-and-Excitation Networks. Thus, without a significant increase in the number of operations and parameters, we add variance pooling and power average pooling to both channel and spatial attention modules.

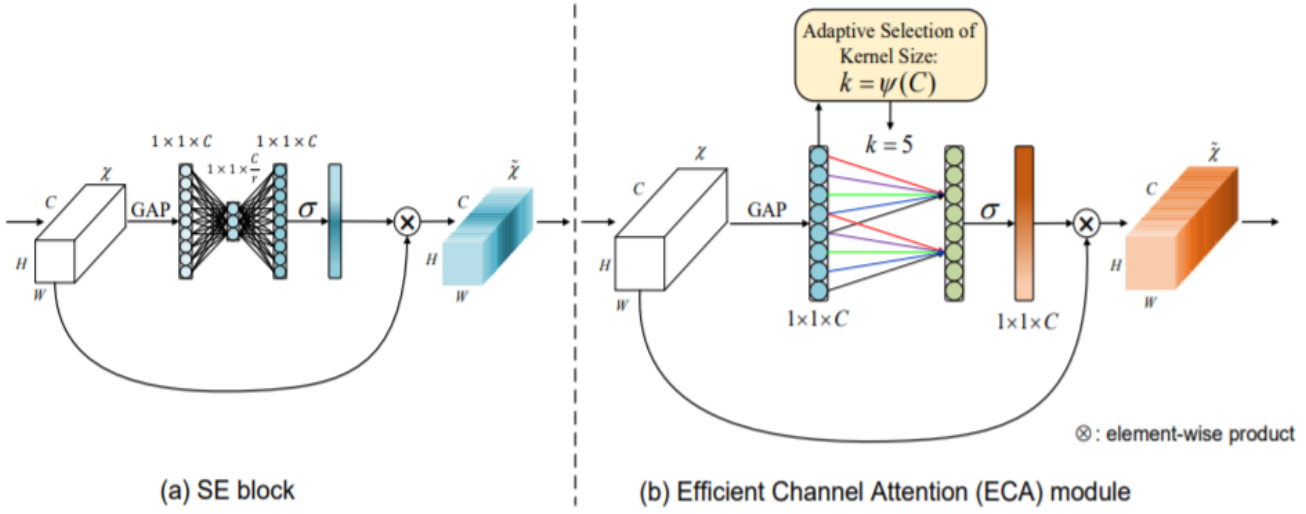

Figure 3: Comparison of (a) SE block and (b) our efficient channel attention (ECA) module.[3]

3 Methodology

Our approach consisted of ablation studies between a baseline ResNet50, a ResNet50 with a CBAM module, and a ResNet50 with our proposed module. We performed these ablation studies with two different datasets, CIFAR-100 and CIFAR-10, resulting in a total of 5 completed different models (all combinations except CIFAR-100 + proposed).

We initially chose the ResNet and EfficientNet[4] architectures because both were state of the art on performance on multiple benchmarks. The versions we chose from each architecture were ResNet-50 and EfficientNet-B0, as these achieved similar performance on the ImageNet dataset, although the earlier has 4.9x more parameters than the later. But, due to resource constraints, only the ResNet-50 models were trained. For the datasets, we chose CIFAR-100 and CIFAR-10 as they are widely used for benchmarking.

3.1 Implementation Details

- **Image augmentation:** We performed Random Horizontal Flipping with a probability of 0,5 and Color Jitter with brightness=0.2, contrast=0.2, saturation=0.2 and hue=0.2,
- **Network modifications:** On ResNet-50, the CBAM module was inserted at the end of every ResBlock. For our proposed module, we modified the original CBAM and added the two extra pooling layers and removed the MPL by the 1D convolution from ECA.
- **Loss function:** The Cross-Entropy loss function with the Softmax classifier was used.
- **Dataset:** CIFAR-100 consists of 100 classes, each having 500 images in the train set and 100 images in the test set. Whereas, CIFAR-10 consists of 10 classes, each having 5000 images in the train set and 1000 images in the test set.
- **Metrics:** We track the Top-1 Acc scores.
- **Optimizer:** Mini-Batch Gradient Descent with Momentum set to 0.9 was used. Momentum is a method that helps accelerate SGD in the relevant direction and dampens oscillations.
- **Learning Rate:** For the learning rate (LR), we used a scheduler to Reduce the LR when the validation accuracy plateaus. We used patience of 6 epochs, a factor of 0.5, and an initial LR of 0.01.
- **Batch size:** We kept a fixed **batch size of 32**. Much research has shown that a mini-batch size of 2-32 yields more stable and generalizable results on multiple benchmarks than large mini-batches.[5]
- **Data-split**: For both CIFAR-10 and CIFAR-100, we performed 85-15 train-validation split on the 50,000 training images. Thus, we trained on 42,500, validated on 7,500, and tested on 10,000 images for each dataset.
- **Image resizing:** For CIFAR-100, we resized the images to 224x224. Whereas, for CIFAR-10, the images are left as 32x32 due to the duration for a single epoch. Because of this, we obtain higher accuracy scores on the tougher CIFAR-100 dataset compared to CIFAR-10, as seen in the table below.
- **Epochs:** For CIFAR-100, we ran the model for 107 epochs; this allowed for convergence in the performance of the models. However, for CIFAR-10, we only trained for 50 epochs due to computing resources. But, CIFAR-10 being a simpler dataset, requires fewer epochs compared to CIFAR-100 to converge.

4 Results

Architecture	Dataset	Validation Acc.	Test Acc.	Param.
ResNet50	CIFAR-10	0.6325	0.64	23.71M
ResNet50 + CBAM		0.6312	0.63	26.24M
ResNet50 + proposed		0.6564	**0.66**	**23.71M**
ResNet50	CIFAR-100	0.7655	0.7677	23.71M
ResNet50 + CBAM		0.7563	0.7737	26.24M
ResNet50 + proposed		-	-	23.71M

Table 1: **Classification results - Top-1 Acc.** on CIFAR-10 and CIFAR-100

Table 1 lists all the deep learning architectures used in the project with the two datasets: CIFAR-100 and CIFAR-10. We were unable to run the proposed variant on CIFAR-100 due to computing resource exhaustion. However, some trends can still be observed. These are:

1. ResNet50 + proposed was the best performing model on CIFAR-10.
2. ResNet50 + proposed has similar number of parameters to vanilla ResNet50 (baseline). Only a difference of 3512 parameters to be precise.
3. Surprisingly, ResNet50 + CBAM performed on-par/slightly worse than vanilla ResNet50. We attribute this difference due to not training it for more epochs. As the CBAM module has 2.5 million more parameters, i.e., higher model capacity. This might require more number of epochs to converge than the other two variants.
4. We see that even though CIFAR-100 is a more difficult dataset (100 classes, each having only 500 images) to perform compared to CIFAR-10 (10 classes, each having 5000 images), we have better accuracy scores. This is because the images were resized to 224x224 for CIFAR-100 and left as 32x32 for CIFAR-10. This shows that the data we feed into these different variants of ResNet50 has a more pronounced effect on the final accuracy score.
5. For CIFAR-100, vanilla ResNet50 performs almost 1% better than with the CBAM module on the validation set. However, on the test images, we see that the CBAM variant performs better than the baseline by 0.6%.

5 Conclusion

In this project, we have compared the performance of vanilla ResNet50, ResNet50 with CBAM module, and a novel attention module, which combines the ideas of both CBAM and ECANet to achieve both spatial and channel-based attention with low model complexity. We were able to show that the proposed attention module achieve considerable performance improvement while keeping the overhead small. We also incorporated two more pool types, variance pooling, and power average pooling, to both the spatial and channel attention.

References

[1] Sanghyun Woo, Jongchan Park, Joon-Young Lee, In So Kweon, "Cbam: Convolutional block attention module," 2018.

[2] Kaiming He Et al., "Deep residual learning for image recognition," 2015.

[3] P. Z. P. L. W. Z. Q. H. Qilong Wang, Banggu Wu, "Eca-net: Efficient channel attention for deep convolutional neural networks," 2019.

[4] Mingxing Tan, Quoc V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019.

[5] C. L. Dominic Masters, "Revisiting small batch training for deep neural networks," 2018.