Score-based Aggregation for Attention Modules in Image Classification Tasks

Changwoo Lee

Department of Electronic Engineering
Hanyang University
Seoul, Korea
cwl30@hanyang.ac.kr

Ki-Seok Chung

Department of Electronic Engineering
Hanyang University
Seoul, Korea
kchung@hanyang.ac.kr

***Abstract*—Deep Convolutional Neural Networks (CNNs) have been widely used for various computer vision tasks because they hierarchically extract bountiful features from a high-dimensional image. Also, some CNNs incorporate channel attention mechanisms that re-scale each channel of intermediate feature maps based on their importance. The channel attention modules squeeze the spatial information of a feature into a representative value to transform it as a re-scaling value. In order to reduce the amount of information, attention modules have utilized hand-designed pooling functions such as max pooling or average pooling which have been widely adopted in CNNs, because they add negligible computational complexity. However, a significant amount of spatial information is lost due to these pooling functions. In this paper, we propose a generalized pooling function that scales down spatial information with respect to the importance of each pixel. Unlike max pooling or average pooling, our score-based aggregation is capable of flexibly adjusting to input. Also, the score-based aggregation function learns how to squeeze the spatial information into the most appropriate representative value, which will convert the pooling into a spatial attention mechanism. Finally, we propose a novel method called Score-based Aggregated Attention Module (SAAM) that utilizes the proposed score-based aggregation. Our experimental results on CIFAR-10 and CIFAR-100 datasets demonstrate that SAAM achieves the highest classification accuracy improvement among existing channel attention modules since the score-based aggregation in SAAM is a more dynamic and effective method than the hand-designed aggregations.**

I. Introduction

Convolutional Neural Networks (CNNs) have unquestionably increased the visual classification performance because they convey rich spatial features from a high-dimensional image [1]–[5]. To express sufficient hierarchical features of an image, deep CNNs consist of a stack of many convolution layers. Also, each convolution layer uses multiple trainable filters to extract multiple features from its input, and the extracted features are considered to be mutually independent of each other along the channel axis. The feature map is formed in a three-dimensional shape—height, width, and channel—and becomes an input of the next convolution layer which is followed by normalization and non-linearities.

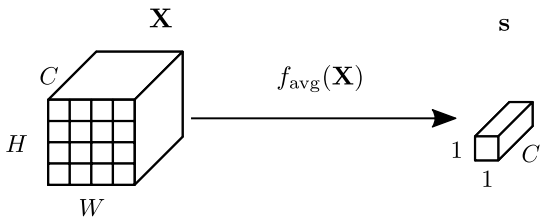
Recent approaches focus on the fact that channels of features should be re-scaled with respect to their importance on a given input image: the process is also known as channel attention. Squeeze-and-Excitation (SE) networks [6] have shown that re-calibrating each channel of a feature can significantly increase the representative power of CNNs. To be specific, an SE network first extracts a scalar value from the spatial domain by channel by aggregating spatial information. Then the SE network transforms the squeezed channel descriptor by a neural network called Multilayer Perceptron (MLP) followed by a sigmoid activation function. The final output of the SE network and the original feature map are multiplied to re-scale it by channel.

In SE networks, the aggregation function is the global average pooling, which means that the spatial information of a channel is reduced to a scalar value by computing the average. In Convolutional Bottleneck Attention Module (CBAM) [7], not only the global average pooling but also the global max pooling is used for the aggregation function, resulting in a dual-path SE-like network. These aggregation methods could be inappropriate because they deliver little spatial information. Moreover, there is a chance to be a better pooling functions for the given tasks. See Fig. 1(a) and 1(b).

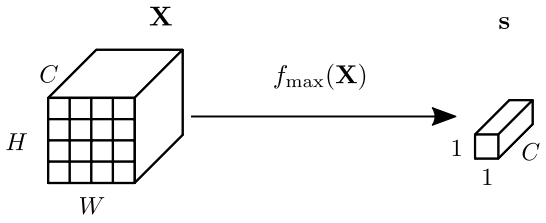
Hence, our work in this paper is motivated by the fact that those methods can be generalized. Correspondingly, we propose a generalized aggregation function in this paper. In the proposed aggregation function, the spatial information is computed by adding pixel-wise predictive *scores* that are outputs of a softmax function followed by a small convolution layer. See Fig. 1(c). Therefore, we call an attention module that uses the score-based aggregation as *Score-based Aggregated Attention Module* (SAAM). The score-based aggregation by a softmax function can generalize the average pooling and the max pooling function. For instance, if one element of the softmax output is close to one while the others are close to zero, the output is a relaxed version of a max pooling function.

One interesting fact is that the pixel-wise scores can hold spatial information because the scores are computed from a small convolution layer, and the scores are given with respect to the importance of pixels. In other words, pooling with pixel-wise scores corresponds to a spatial attention.

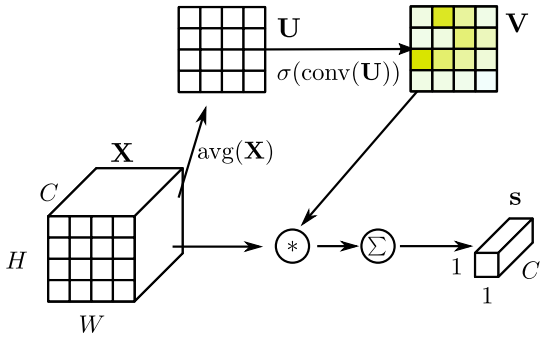
For experiments, we first conduct an ablation study to verify the effectiveness of SAAM on CIFAR-10 and CIFAR-100 [8]. By changing the number of paths in SAAM, the exper-



(a) Average pooling used in SE [6] and CBAM [7].



(b) Max pooling used in CBAM [7].



(c) Score-based aggregation in SAAM

Fig. 1: Different types of pooling for spatial feature aggregation in attention modules. Our proposed score-based pooling (c) can generalize both the average pooling (a) and the max pooling (b). In (c), $\sigma(\cdot)$ denotes the softmax activation.

imental results show that the single path SAAM surpasses the multi-path SAAM because the score-based aggregation behaves like a spatial attention mechanism. Also, we validate the performance of SAAM with the state-of-the-art attention modules—SE [6], SE with both average pooling and max pooling, and CBAM [7]. We plug SAAM, SE, and CBAM in ResNet [4] with various depths. The experimental results show that SAAM significantly outperforms the previously proposed attention modules.

II. Related Works

A. Attention Mechanisms in CNN

The attention mechanism has emerged as a method to solve problems in a field called neural machine translation by giving more weights to essential words [9]–[12]. Recently, the idea of the attention model has been adopted to enhance the representative power of CNNs. In *Squeeze-and-Excitation* (SE) network [6], the channel information of an intermediate feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ is dynamically re-calibrated by a vector $\mathbf{r}$ whose C elements all lie in the range $(0, 1)$. The re-scaling vector $\mathbf{r}$ is computed by two steps: *aggregation* and *transformation*. In an aggregation step, the feature map $\mathbf{X}$ is squeezed into $\mathbf{s} \in \mathbb{R}^C$ by an aggregation function $f : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^C$. An element of $\mathbf{s}$ represents the identity of the corresponding channel of $\mathbf{X}$, by a scalar value. In SE [6], the authors call $\mathbf{s}$ as a channel descriptor. Then, $\mathbf{s}$ is transformed to $\mathbf{r}$ by the function $g : \mathbb{R}^C \to \mathbb{R}^C$, which consists of two sequential fully-connected layers with a ReLU [13] nonlinearity. Finally, $\mathbf{r}$, which is a sigmoid activation, is element-wisely multiplied to $\mathbf{X}$ to achieve the channel attention.

Although SE networks [6] augment the feature representation along the channels of a feature map, they do not preserve any spatial information because they use the global average aggregation that the spatial information is shrunk into a scalar value as the mean of each pixel. This type of aggregation can be treated as a "static" aggregation since the weight per pixel for aggregation does not change (The details will be described in Section III-A). In [7], the authors proposed a method called Convolutional Bottleneck Attention Module (CBAM) claiming that using not only the global average pooling but also the global max pooling improves the overall performance of channel attention. The experimental results in [7] show that using multiple types of pooling methods is better than using an aggregation, but only a few static aggregation methods are available—average pooling, max pooling, and min pooling, for example. One can also rebut that those static hand-designed aggregation methods might not be optimal, even though multiple of them are in an attention module. However, our proposed aggregation method is not only flexible but can be trained to be optimal. On top of that, CBAM utilizes a spatial attention mechanism as well. The spatial attention module, which is followed by a channel attention module, re-calibrates each pixel by a sigmoid activation of an output of a convolution layer, by squeezing the channel information with the global max pooling and the global average pooling. We argue in Section V-B2 that a softmax activation is more appropriate than a sigmoid activation.

B. Dynamic Pooling

The static poolings have been commonly used in CNNs, but the limitation of them is obvious: they do not preserve the spatial information. Hence there have been some efforts to minimize the information loss with dynamic poolings.

The stochastic pooling [14] randomly samples one of the pixels in a pooling window with respect to the softmax probability of its input. Unlike our method, the stochastic pooling lacks scalability because it requires extensive random number generations.

Zhang et al. [15] introduced a global pooling layer, which consists of log-mean-exp function. They use a trainable value for the log-mean-exp function, but the value does not vary from instance to instance, while ours can be altered. Gridhar and Ramanan [16] proposed an attention-based pooling for human action recognition tasks. Although their approach is based on the attention mechanism, they use an attention model which is based on class-specific weights (weights per class)

since they place the dynamic pooling layer at the end of the CNN (right before the linear classifier). However, our method does not need any class-specific weights.

III. The Proposed Method

In this paper, we generalize an aggregation function which is commonly used for shrinking the high-dimensional spatial information into the low-dimensional space to have a channel descriptor. We first introduce how the commonly used aggregation methods, such as max pooling and average pooling, can be generalized by a score-based aggregation. Then we present how the score-based aggregation can be adopted to the attention module.

A. Score-based Aggregation

We explore the identity of the global average pooling and the global max pooling, which are commonly used for aggregating the spatial information in CNNs. Given an input tensor $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, the global average pooling function $f_{\text{avg}} : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^C$ is defined as the following formula:

$$\begin{aligned} s_c &= f_{\text{avg}}(\mathbf{X}_c) \\ &= \frac{1}{HW}\sum_h\sum_w x_{h,w,c} \\ &= \sum_h\sum_w \frac{1}{HW} x_{h,w,c} \\ &= \sum_h\sum_w v^{\text{avg}} x_{h,w,c}, \end{aligned} \tag{1}$$

where

$$v^{\text{avg}} = \frac{1}{HW}$$

and $\mathbf{X}_c$ and $\hat{\mathbf{X}}_c$ are the c-th channel of the input and output of f_{avg}, respectively. Similarly, the global max pooling function $f_{\max} : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^C$ is:

$$\begin{aligned} s_c &= f_{\max}(\mathbf{X}_c) \\ &= \sum_h\sum_w v^{\max}_{h,w} x_{h,w,c}, \end{aligned} \tag{2}$$

where

$$v^{\max}_{h',w'} = \begin{cases} 1 & \text{if } x_{h',w',c} \geq x_{h,w,c},\ \forall h, w \\ 0 & \text{otherwise.} \end{cases}$$

These aggregation methods are based on *static* and *hand-designed* weights, as we mentioned in Section II-A. Also, they are too biased: average poolings assign equal weights for all pixels without considering the importance, and max poolings neglect all but one pixel. With this view, it is hard to say that the outputs of these functions effectively represent the original input. To remodel the static aggregation as a flexible and generalized one, we focus on the properties of $\mathbf{V}^{\max}$ and $\mathbf{V}^{\text{avg}}$. Given a $H\times W$ spatial weight matrix $\mathbf{V}$ whose elements have the following properties:

$$0 \leq v_{h,w} \leq 1, \forall h, w, \quad \sum_h\sum_w v_{h,w} = 1, \tag{3}$$

TABLE I. Steps in Score-based Aggregated Attention Module (SAAM)

Step	Operation (Eq. #)	Output notation (shape)
1	Channel aggregation (5)	$\mathbf{U}(H\times W\times 1)$
2	Convolution (6)	$\tilde{\mathbf{U}}(H\times W\times 1)$
3	Softmax (7)	$\mathbf{V}(H\times W\times 1)$
4	Spatial aggregation (4)	$\mathbf{s}(1\times 1\times C)$
5	Transformation (8), (9)	$\mathbf{r}(1\times 1\times C)$
6	Channel-wise recalibration (10)	$\hat{\mathbf{X}}(H\times W\times C)$

we can generalize the global average pooling and the global max pooling by $\mathbf{V}$ because f_{avg} and $f_{\max}$ are particular cases of the weighted sum of the input tensor $\mathbf{X}$ by $\mathbf{V}$. Each element of the spatial weight $\mathbf{V}$ can be regarded as a score per spatial location of $\mathbf{X}$. Hence, we call the generalized pooling function as *score-based aggregation*. The *score-based aggregation* function $f : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^C$ is then:

$$\begin{aligned} s_c &= f(\mathbf{X}_c) \\ &= \sum_h\sum_w v_{h,w} x_{h,w,c}. \end{aligned} \tag{4}$$

We further extend the score-based aggregation by adopting a spatial attention mechanism to the spatial score matrix $\mathbf{V}$. That is, if $\mathbf{V}$ becomes input-specific, and it can be trained to contain useful spatial information, we can minimize the spatial information loss while aggregating.

To generate the input-specific spatial weight $\mathbf{V}$, we follow three steps: First, we squeeze the information of the input tensor $\mathbf{X}$ along the channel dimension by averaging to have $\mathbf{U} \in \mathbb{R}^{H\times W}$:

$$\mathbf{U} = \frac{1}{C}\sum_c \mathbf{X}_c. \tag{5}$$

Second, we transform the remaining spatial information with a small convolution layer $\text{Conv} : \mathbb{R}^{H\times W\times 1} \to \mathbb{R}^{H\times W\times 1}$:

$$\tilde{\mathbf{U}} = \text{Conv}(\mathbf{U}). \tag{6}$$

We now recall the properties of $\mathbf{V}$ in (3): every element in $\mathbf{V}$ is a value in range $(0, 1)$ and the sum of all elements of $\mathbf{V}$ is 1. The score matrix $\mathbf{V}$ is normalized by softmax activations since the output of the softmax function shares the same property in (3). Therefore, the score matrix $\mathbf{V}$ which will be used in (4) can be achieved from a softmax output of $\tilde{\mathbf{U}}$:

$$v_{h,w} = \frac{\exp(\tilde{u}_{h,w})}{\sum_{h'}\sum_{w'}\exp(\tilde{u}_{h',w'})}. \tag{7}$$

One can quickly notice that $\mathbf{V}$ is a probability distribution of which pixel is going to be selected as an output of the pooling function, because of the properties in (3). Hence, unlike the hand-designed pooling, our method gives trainable and input-aware scores, which lie somewhere between constant equal weights (average pooling) and extreme, one-or-nothing weights (max pooling), concerning the importance of each pixel.

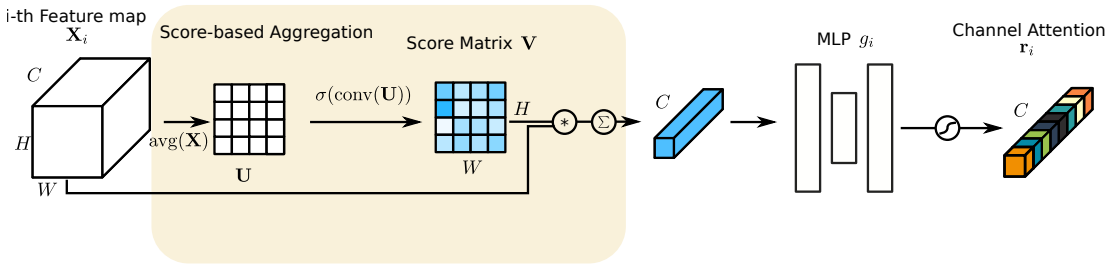


Fig. 2: Score-based Aggregated Attention Module.

B. Score-based Aggregation in Channel Attention Modules

The output of the score-based aggregation function is fed into a transformation function g:

$$\mathbf{r} = \text{sigmoid}\big(g(f(\mathbf{X})\big) \tag{8}$$

where $\text{sigmoid}(x) = \frac{1}{1+\exp(-x)}$ and g consists of two fully-connected layers and a ReLU nonlinearity ϕ:

$$g(f(\mathbf{X})) = W_2\phi(W_1 f(\mathbf{X})). \tag{9}$$

In (9), W_1 and W_2 are the weight matrices of dense layers. The final result of the channel attention module $\mathbf{r}$ and the feature map $\mathbf{X_c}$ are channel-wise multiplied with each other to obtain $\hat{\mathbf{X}}_\mathbf{c}$:

$$\hat{\mathbf{X}}_c = r_c \mathbf{X}_c. \tag{10}$$

$\hat{\mathbf{X}}_\mathbf{c}$ is then fed into the next convolution layer. For brevity, we will call the channel attention module with the score-based aggregation as *Score-based Aggregated Attention Module* (SAAM). The overall procedure of SAAM is summarized in Table I and Fig. 2.

One advantage of SAAM that the number of paths in the attention module is not limited by the number of static pooling methods. In CBAM [7], there are two paths, one from the global average and the other from the global max pooling, and they share the transformation function g. One can add more paths by putting other static pooling methods such as min pooling, but the number of static poolings is limited. However, because SAAM employs the generalized aggregation function, we can set the number of paths by implementing multiple score-based aggregations.

We suggest that using one path in SAAM is good enough, even though it is counter-intuitive. Because the score-base aggregation already reduces the spatial information loss, the multi-path attention module is no longer needed. We will present experimental results and analyze results in Section V-A.

IV. Implementation

A. Base Network

ResNet [4] is employed as a base CNN architecture. Resnet with various depths are chosen—specifically, ResNet-20, ResNet-56, and ResNet-110, where the suffix represents the total number of convolution layers and fully-connected layers.

B. Attention Modules

During the transformation stage of SE [6], CBAM [7] and SAAM, the dimension is reduced in order to reduce the amount of memory requirement and that of computations. For example, when the dimension of $\mathbf{s} = f_{\text{avg}}(\mathbf{X})$ (the squeezed information by the aggregation function f_{avg}) is C, the dimension of $W_1\mathbf{s}$ in (9) is reduced to $\frac{C}{t}$ where $t > 1$. For all networks, we set t to be 4. That is, the dimension of $\mathbf{s}$ is once reduced to $\frac{C}{4}$. Following SE networks, all channel attention modules are applied before summation with the identity branch of ResNet [6]. Also, we use a 5×5 convolution layer for SAAM as in (6). It increases a negligible amount of computation since the number of channels in the input and that in the output of the convolution layer are both 1.

C. Notations

We implement SAAM with 1, 2, and 4 paths and mark as SAAM-K-D where K and D denote the number of paths and that of layers in CNN, respectively. For example, SAAM-4-20 represents SAAM with 4 paths, which is attached to ResNet-20. Also, we will refer SE-1-D as a SE network [6] with the global average pooling (single path), and SE-2-D as a SE network with both the global average pooling and the global max pooling (a dual-path CBAM [7] style). Finally, we will denote the baseline model as ResNet-D.

D. Overheads

The trade-off between the performance and the memory and computational overhead is significant to deploy the neural network in practice. We analyze the memory and the computational requirements of ResNet-20, SE-1-20, CBAM-2-20, and SAAM-1-20. ResNet-20 requires 280K parameters and 43.14 MFLOPs to classify a $32 \times 32 \times 3$ CIFAR-100 image. SE-1-20 requires 288K parameters and 43.15 MFLOPs. Also, CBAM-2-20 requires 288.7K parameters and 43.35 MFLOPS, which the number of parameters and computations are 3.04% and 0.49% larger than those of ResNet-20, respectively. The number of parameters and computations of SAAM are in between SE-1-20 and CBAM-2-20. SAAM-1-20 requires 288.5K parameters and 43.25 MFLOPs, corresponding to 2.96% and 0.25% increase over ResNet-20. Since the complexity of the memory and computations increased by SAAM are negligible to that of original ResNet, the performance gain per increased parameters and computations are reasonable.

E. Training

We train all networks on CIFAR-10 and CIFAR-100 [8] datasets with 182 epochs. Both datasets contain 60,000 32×32 color images with 10 and 100 classes, respectively. We take 50,000 images for training and 10,000 images for testing on both datasets. The training data are augmented with random shuffling, random flipping, and random cropping. Although CIFAR-100 has ten coarse classes, we use 100 fine-grained classes for training and verification. We apply l2 regularization loss with a multiple of 0.0002 to all parameters except the attention modules. A widely used method called Stochastic Gradient Descent with a momentum of 0.9 is used for all networks. Also, the learning rate starts from 0.1 and decays to 0.01 and 0.001 at the 91^{st} and 136^{th} epoch, respectively.

V. Experiments

We conduct experiments to examine the performance of SAAM. In Section V-A, we analyze the relation between the number of paths in SAAM and the performance. In Section V-B, ResNets with SAAM are compared to ResNets with the state-of-the-art attention modules with static aggregation—SE [6] and SE with both average pooling and max pooling. Also, we analyze SAAM with CBAM to explore the effectiveness of the spatial attention mechanism of SAAM.

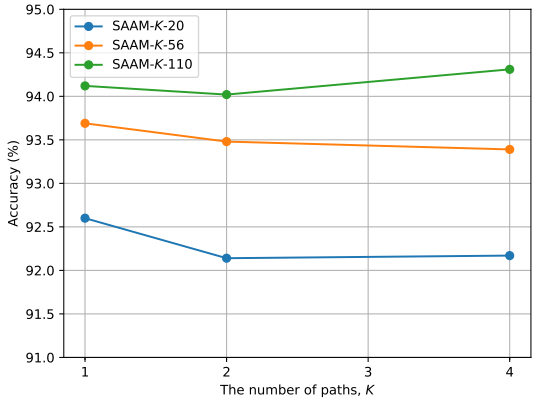
A. Ablation Study: Single-path vs. Multi-path SAAM

Since SAAM can have a unlimited number of paths, we examine how the accuracy of SAAM changes with respect to the number of paths. In Fig. 3, the relation between the accuracy and the number of paths is presented. The results show that the multi-path SAAM does not always achieves a higher accuracy than the single-path SAAM. Instead, the accuracy tends to decrease when the number of paths is increased. The accuracy of SAAM-1-56 is 72.49% on CIFAR-100, but the accuracy of SAAM-2-56 and SAAM-4-56 are 71.48% and 71.30%, respectively. Although SAAM-4-110 and SAAM-4-20 show the best accuracy on CIFAR-10 and CIFAR-100, respectively, using the single path SAAM may be preferred because the performance difference is negligible yet the computational overhead for 4-path SAAMs is quadrupled.

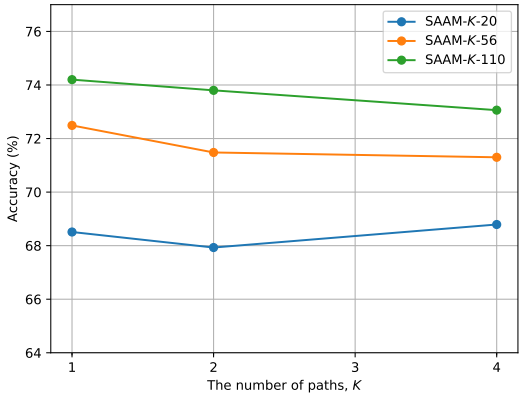
We suppose that this is because the score-based aggregation captures rich spatial information; hence, it does not need additional aggregation functions. In contrast, CBAM uses the global max pooling and the global average pooling for the aggregation functions: they cannot compress the spatial information individually, but using both altogether can preserve spatial information to some extent. In summary, the experimental results show that the single path SAAM is good enough to show the effectiveness of the score-based aggregation.

B. Performance Comparison

We examine the classification accuracy of the baseline ResNets, SE networks [6], CBAM [7], and SAAM on CIFAR-10 and CIFAR-100. We selected SAAMs with a single path because the single path SAAMs showed the best accuracy at the majority of experiments in Section V-A. First, the



(a) CIFAR-10



(b) CIFAR-100

Fig. 3: Accuracy of SAAM with different number of paths on CIFAR-10 and CIFAR-100. Multiple paths do not guarantee the high performance.

TABLE II. Top-1 accuracy on CIFAR-10 (%).

Attention Type	K	ResNet Size		
		20	56	110
No attention	N/A	91.97	92.90	93.60
SE [6] (AvgPool)	1	92.04	93.66	94.06
SE [6] (AvgPool & MaxPool)	2	91.81	93.4	93.91
CBAM [7]	2	91.78	93.20	93.82
SAAM (ours)	1	**92.60**	**93.69**	**94.12**

TABLE III. Top-1 accuracy on CIFAR-100 (%)

Attention Type	K	ResNet Size		
		20	56	110
No attention	N/A	67.43	71.33	72.83
SE [6] (AvgPool)	1	68.06	71.78	73.76
SE [6] (AvgPool & MaxPool)	2	68.10	71.88	73.71
CBAM [7]	2	67.84	72.16	73.24
SAAM (ours)	1	**68.51**	**72.49**	**74.20**

performance comparison of SAAM and two types of SE networks [6] is presented. We next compare SAAM with the other spatial attention module, CBAM [7]. The experimental results on CIFAR-10 and CIFAR-100 are summarized in Table II and Table III, respectively.

1) SAAM vs. Channel Attention Modules: We analyze the performance of SAAM with SE networks to examine whether the score-based pooling is competent enough when compared with the static hand-designed pooling methods. Therefore, we train two SE networks—one uses the global average pooling as an aggregation function, and the other uses both the global average pooling and the global max pooling. Note that the latter is equal to CBAM [7] without any spatial attention module.

On CIFAR-10 dataset, SAAM-1-20, SAAM-1-56, and SAAM-1-110 outperform both the single path SE and the dual path SE networks. The accuracy of SAAM-1-20 is 92.60% (an error rate of 0.0740), which is 7.04% lower error rate than that of SE-1-20. Surprisingly, the accuracy of SAAM-1-56 is even higher than the accuracy of ResNet-110, although the number of layers of SAAM-1-56 is almost half that of ResNet-110.

The results are similar on CIFAR-100 dataset, which is more challenging than CIFAR-10. SAAM outperforms the other channel attention modules with large margins. For example, the accuracy of SAAM-1-110 is 74.20%, and the error rate of it is 1.68% lower than that of SE-1-110. The experimental results show that the score-based aggregation compresses better spatial features than the global average pooling or the global max pooling.

2) SAAM vs. Spatial Attention Module: Because SAAM utilizes spatial attention while aggregating the spatial information, we compare the performance of SAAM with the state-of-the-art spatial attention module, CBAM [7], which is another variant of SE [6]. The major architectural difference between SAAM and CBAM is that the spatial attention of SAAM is done by a softmax output, which is a generalization of the routinely used aggregation functions, while that of CBAM is basically an output of the sigmoid function.

We conduct experiments on CIFAR-10 and CIFAR-100. The experimental results show that SAAMs outplays CBAM with a less amount of computation and a fewer number of weights. The accuracy of SAAM-1-20, SAAM-1-56, and SAAM-1-110 is higher than that of CBAM-2-20, CBAM-2-56, and CBAM-110, respectively. We explain the results with respect to the different functions which are used for spatial attention in SAAM and CBAM, as we stated above. Because the softmax outputs are Categorical distributions over a spatial dimension, the softmax function is more appropriate to spatial attention than sigmoid outputs, which represent Bernoulli distributions over individual pixel.

VI. Conclusion and Future Works

In this paper, we investigate the problem of aggregation methods in conventional channel attention modules and minimize the spatial information loss via the score-based aggregation, which is a generalized pooling method. Because the score of each pixel represents the importance of the pixel, the spatial information is effectively preserved. Without adding too many redundant parameters and computations, the experimental results on CIFAR-10 and CIFAR-100 show that our proposed method achieved superlative accuracy improvement compared with other state-of-the-art channel attention modules. We will explore other applications that suffer from information loss while aggregating in future work. Also, since the very first aggregation along the channel axis in the proposed method is still static aggregation method, we will change this average pooling with more appropriate aggregation methods.

Acknowledgment

This research was funded by the Technology Innovation Program MOTIE (No. 10076583), the Competency Development Program for Industry Specialists MOTIE (No. 0001883), and IC Design Education Center (IDEC), Korea.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

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

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[7] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[8] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[9] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[11] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-9)*, 2010, pp. 807–814.

[14] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[15] B. Zhang, Q. Zhao, W. Feng, and S. Lyu, "Alphamex: A smarter global pooling method for convolutional neural networks," *Neurocomputing*, vol. 321, pp. 36–48, 2018.

[16] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Advances in Neural Information Processing Systems*, 2017, pp. 34–45.