RFAConv: Innovating Spatial Attention and Standard Convolutional Operation

Xin Zhang[1], Chen Liu[1], Tingting Song[1,*], Degang Yang[1,2,*], Yichen Ye[3], Ke Li[1], and Yingze Song[1]

[1] College of Computer and Information Science, Chongqing Normal University
[2] Chongqing Engineering Research Center of Educational Big Data Intelligent Perception and Application
[3] College of Electronic and Information Engineering, Southwest University
*Corresponding author: E-mail:address:{ttsong,yangdg}@cqnu.edu.cn

Abstract. Spatial attention has been widely used to improve the performance of convolutional neural networks. However, it has certain limitations. In this paper, we propose a new perspective on the effectiveness of spatial attention, which is that the spatial attention mechanism essentially solves the problem of convolutional kernel parameter sharing. However, the information contained in the attention map generated by spatial attention is not sufficient for large-size convolutional kernels. Therefore, we propose a novel attention mechanism called Receptive-Field Attention (RFA). Existing spatial attention, such as Convolutional Block Attention Module (CBAM) and Coordinated Attention (CA) focus only on spatial features, which does not fully address the problem of convolutional kernel parameter sharing. In contrast, RFA not only focuses on the receptive-field spatial feature but also provides effective attention weights for large-size convolutional kernels. The Receptive-Field Attention convolutional operation (RFAConv), developed by RFA, represents a new approach to replace the standard convolution operation. It offers nearly negligible increment of computational cost and parameters, while significantly improving network performance. We conducted a series of experiments on ImageNet-1k, COCO, and VOC datasets to demonstrate the superiority of our approach. Of particular importance, we believe that it is time to shift focus from spatial features to receptive-field spatial features for current spatial attention mechanisms. In this way, we can further improve network performance and achieve even better results. The code and pre-trained models for the relevant tasks can be found at https://github.com/Liuchen1997/RFAConv.

1 Introduction

Convolutional neural networks [1, 2] have dramatically reduced the computational overhead and complexity of models by using the convolutional operation with shared parameters. Driven by classical networks, such as LeNet [3], AlexNet [4], and VGG [5], convolutional neural networks have now established a complete system and formed advanced convolutional neural network models [6, 7, 8, 9, 10]. After carefully studying the convolutional operation, we gained inspiration. For classification, object detection, and semantic segmentation tasks, on one hand, the shape, size, color, and distribution of objects in different locations in the image are variable. However, during the convolutional operation, the convolutional kernel uses the same parameters in each receptive field to extract information, which does not consider the differential information from different locations. So the performance of the network is limited, as demonstrated by recent works [11, 12, 13]. On the other hand, the convolutional process does not take into account the significance of each feature, which further reduces the efficiency of the extraction features and ultimately restricts the performance of the model. In addition, the attention mechanism [14, 15, 16] enables the model to concentrate on significant features, which can enhance the benefits of feature extraction and the ability of convolutional neural networks to capture detailed feature information.

By examining the intrinsic limitations of convolutional operations and the properties of attention mechanisms, we assert that while the current spatial attention mechanism has fundamentally addressed the issue of parameter sharing in convolutional operations, but it remains restricted to the recognition of spatial features. The current spatial attention mechanism does not fully address the parameter sharing problem for larger convolutional kernels. Furthermore, they are unable to emphasize the significance of each feature in the receptive field, such as the existing Convolutional Block Attention Module (CBAM) [17] and Coordinate Attention (CA) [18]. Consequently, we propose a novel receptive-field attention (RFA) that comprehensively addresses the issue of parameter sharing for convolutional kernels and takes into account the significance of each feature in the receptive field. The RFA-designed convolutional operation (RFAConv) is a groundbreaking method that can replace standard convolutional operations in current neural networks. With only a few additional parameters and computational overhead, RFAConv enhances network performance. Numerous experiments conduct on ImagNet-1k [19], COCO [20], and VOC [21] have demonstrated the efficacy of RFAConv. As an attention-based convolutional operation, RFAConv outperforms CAMConv, CBAMConv, CAConv (constructed by CAM [17], CBAM, and CA, respectively), as well as the standard convolutional operation. Moreover, to address the issue of slow to extract receptive field features for current methods, we propose a lightweight operation. During the construction of RFAConv, we also design an upgraded version of CBAM and CA and conduct relevant experiments. We assert that spatial attention mechanisms should focus on receptive-field spatial features to further advance their development and enhance the advantages of convolutional neural networks.

2 Related Works

2.1 Convolutional neural network architecture

The convolutional operation, which serves as a basic operation in convolutional neural networks, has led the development of many advanced network mode, such as vehicle detection [22], UAV images [23], medicine [24], etc. He et al. [25] suggested that as the depth of the network increases, the model becomes harder to train and may experience a degradation phenomenon. To address this issue, they proposed to use residual connections to revolutionize the design of network. Huang et al. [26] improved feature information by reusing features to address the issue of network gradient disappearance. After conducting a thorough study on convolutional operations, Dai et al. [27] claimed that the convolutional operation

with a fixed sampling position can restrict the performance of the network to a certain extent. So, they proposed Deformable Conv, which alters the sampling positions of convolutional kernels by learning offsets. Building upon this approach, Deformable Conv V2 [28], and Deformable Conv V3 [29] were subsequently developed to further enhance the performance of convolutional networks. Zhang et al. [30] observed that group convolution could decrease the number of parameters and computational overhead of the model, however, insufficient interaction between information within the group can adversely affect the final network performance. Although the 1×1 convolution can interact with information, it will bring more parameters and computational overhead, so they proposed the parameter-free Channel Shuffle operation to interact with the information between groups. Ma et al. [31] discovered that models with few parameters do not always result in faster inference times, and similarly, small computational effort does not guarantee quick performance. After careful study, they proposed the ShuffleNet V2. The YOLO [32] object detection network divided the input image into a grid to predict the location and class of objects. As research has progressed, eight versions of object detectors based on YOLO have been proposed, such as YOLOv5 [33], YOLOv7 [34], YOLOv8 [35], etc. While the previously-mentioned convolutional neural network architectures have achieved significant success, they do not directly address the problem of parameter sharing during the feature extraction process. Our work focuses on utilizing the attention mechanism to tackle the problem of convolutional parameter sharing from a fresh perspective.

2.2 Attention Mechanism

Attention mechanism, as a technique to improve the performance of network models, allows models to focus on key features. The theory of attention mechanism has now established a complete and mature system in the field of deep learning. Hu et al. [36] proposed a Squeeze-and-Excitation (SE) block to obtain the weights corresponding to each channel. This is achieved by compressing features to aggregate global channel information. Wang et al. [37] asserted that the correspondence between individual channels and weights is indirect when the SE interacts with information. Therefore, they designed the Efficient Channel Attention (ECA) by replacing the Fully Connected (FC) layer in the SE with a one-dimensional convolution of adaptive kernel size. Woo et al. [17] proposed the Convolutional Block Attention Module (CBAM), which combines channel attention and spatial attention. As a plug-and-play module, it can be embedded into convolutional neural networks to enhance network performance. Although SE and CBAM have allowed the network to achieve good performance, Hou et al. [18] found that the compressing feature in SE and CBAM lost too much information. Therefore, they proposed the lightweight coordinate attention (CA) to solve this problem. Fu et al. [38] designed a spatial attention module and a channel attention module to extend Fully Convolutional Networks (FCN) for modeling semantic interdependencies in the spatial and channel dimensions, respectively. Zhang et al. [39] generated feature maps at different scales on channels to build a more efficient channel attention mechanism. This paper introduces a new approach to address the issue of parameter sharing in standard convolutional operations. Our proposal involves combining attention mechanisms to create convolutional operations. Although existing attention mechanisms have demonstrated good performance, they do not specifically target the spatial features of receptive fields. To tackle this limitation, we developed RFA-Conv with non-shared parameters to improve performance of the network.

3 Methods

3.1 Reviewing Standard Convolutional Operation

The standard convolutional operation serves as the fundamental building block for constructing convolutional neural networks. It utilizes sliding windows with shared parameters to extract feature information and overcome the inherent issues of neural networks constructed with fully connected layers, such as a large number of parameters and high computational overhead. Let $X \in R^{C\times H\times W}$ denote the input feature maps, where C, H, and W represent the number of channels, height, and width of the feature map, respectively. To clearly demonstrate the feature extraction process through convolutional kernels, we use the example of C = 1. The convolutional operation to extract feature information from each receptive field slider can be expressed as follows:

$$\begin{aligned} F_1 &= X_{11}\times K_1 + X_{12}\times K_2 + X_{13}\times K_3 + ... + X_{1S}\times K_S \\ F_2 &= X_{21}\times K_1 + X_{22}\times K_2 + X_{23}\times K_3 + ... + X_{2S}\times K_S \\ &... \\ F_N &= X_{N1}\times K_1 + X_{N2}\times K_2 + X_{N3}\times K_3 + ... + X_{NS}\times K_S \end{aligned} \tag{1}$$

Here, F_i represents the value obtained by each convolutional slider after computation, X_i represents the pixel value at the corresponding position within each slider, K represents the convolutional kernel, S denotes the number of parameters in the convolutional kernel, and N represents the total number of the receptive-field slider. As Shown in Figrue 1, K is a 3×3 convolutional kernel and S is 9. Moreover, there are a total of 16 receptive-field sliders in the Figrue 1. For simplicity of representation, we just draw three receptive-field sliders. It can be seen that features at the same position within each slider share the same parameter K_i. Therefore, the standard convolutional operation fails to capture the difference in information brought by different positions, which ultimately limits the performance of the convolutional neural network to a certain extent.

3.2 Reviewing Spatial Attention

Currently, the spatial attention mechanism uses the attention map obtained through learning to highlight the importance of each feature. Similar to the previous section, C=1 is taken as an example. The spatial attention mechanism highlighting key features can be expressed simply as follows:

$$\begin{aligned} F_1 &= X_1 \times A_1 \\ F_2 &= X_2 \times A_2 \\ &... \\ F_N &= X_N \times A_N \end{aligned} \tag{2}$$

Here, F_i represents the value obtained after the weighting operation. X_i and A_i represent the values of the input feature map and the learned attention map at different positions, respectively, N is the product of the height and width

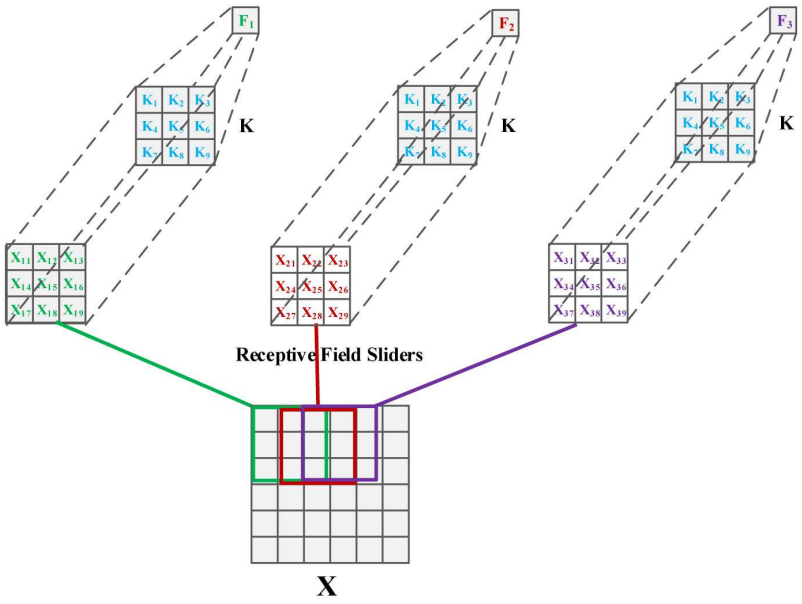

Fig. 1. It simply represents a 3×3 convolution operation. The features are obtained by multiplying the convolution kernel with a receptive-field slider of the same size and then summing.

of the input feature map, which represents the total number of pixel values. In general, the whole process can be simply represented in Fig. 2.

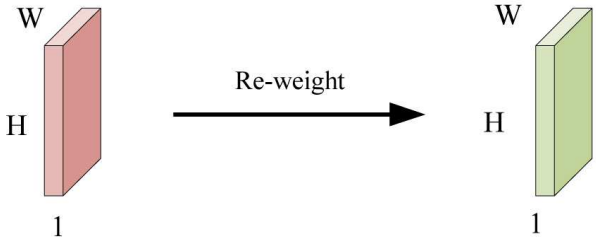

Fig. 2. The original feature map highlights the key features by learned attention map. This process of highlighting is the Re-weight (×) operation.

3.3 Spatial Attention and Standard Convolutional Operation

As it is widely recognized, incorporating attention mechanisms into convolutional neural networks can enhance their performance. After a careful study of the standard convolution operation and the existing spatial attention mechanism, we argue that the spatial attention mechanism effectively overcomes the inherent limitation of convolutional neural networks, which is parameter sharing. Currently, the most commonly used kernel sizes in convolutional neural networks are 1×1 and 3×3. The convolutional operation for extracting features after introducing the spatial attention mechanism is either a 1×1 or a 3×3 convolutional operation. To intuitively show the process, the spatial attention mechanism is inserted into the front of the 1×1 convolutional operation. Through the attention map to weigh operation the input feature map (Re-weight "×"), finally, the slider feature information of the receptive-field is extracted through 1×1 convolution operation. The overall process can be simply represented as follows:

$$\begin{aligned} F_1 &= X_1 \times A_1 \times K \\ F_2 &= X_2 \times A_2 \times K \\ &... \\ F_N &= X_N \times A_N \times K \end{aligned} \tag{3}$$

Here, the convolution kernel K represents only one parameter value. If we take the value of $A_i \times K$ as a new convolution kernel parameter, the interesting thing is that the problem of parameter sharing in the extraction of features by 1×1 convolution operations is solved. As shown in Fig. 3, it visualizes the combination of a 1×1 convolutional kernel and a spatial attention mechanism. This combination essentially solves the problem of parameter sharing. Specifically, the output fea-

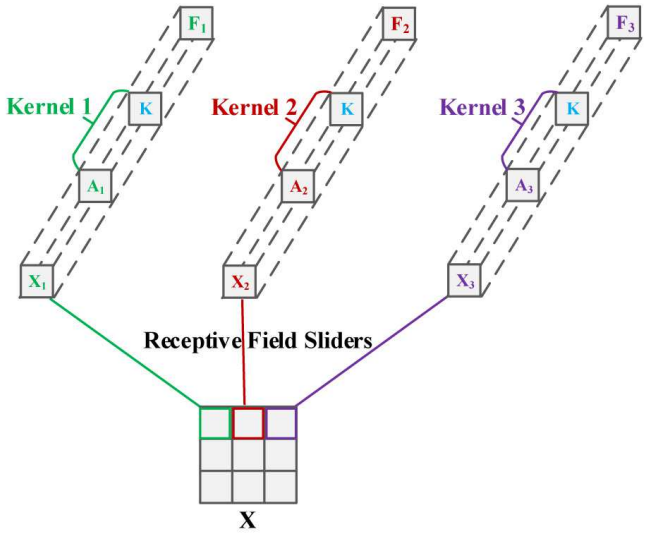

Fig. 3. The convolutional kernel parameter K_i obtained by multiplying the attentional weight A_i with the convolutional kernel parameter K is different in each receptive-field slider, i.e., $Kernel1 \neq Kernel2 \neq Kernel3 \neq ... \neq KernelN$.

$$
\begin{aligned}
F_1 &= X_{11} \times A_{11} \times K_1 + X_{12} \times A_{12} \times K_2 + X_{13} \times A_{13} \times K_3 + ... + X_{19} \times A_{19} \times K_9 \\
F_2 &= X_{21} \times A_{21} \times K_1 + X_{22} \times A_{22} \times K_2 + X_{23} \times A_{23} \times K_3 + ... + X_{29} \times A_{29} \times K_9 \\
&... \\
F_N &= X_{N1} \times A_{N1} \times K_1 + X_{N2} \times A_{N2} \times K_2 + X_{N3} \times A_{N3} \times K_3 + ... + X_{N9} \times A_{N9} \times K_9
\end{aligned}
\tag{4}
$$

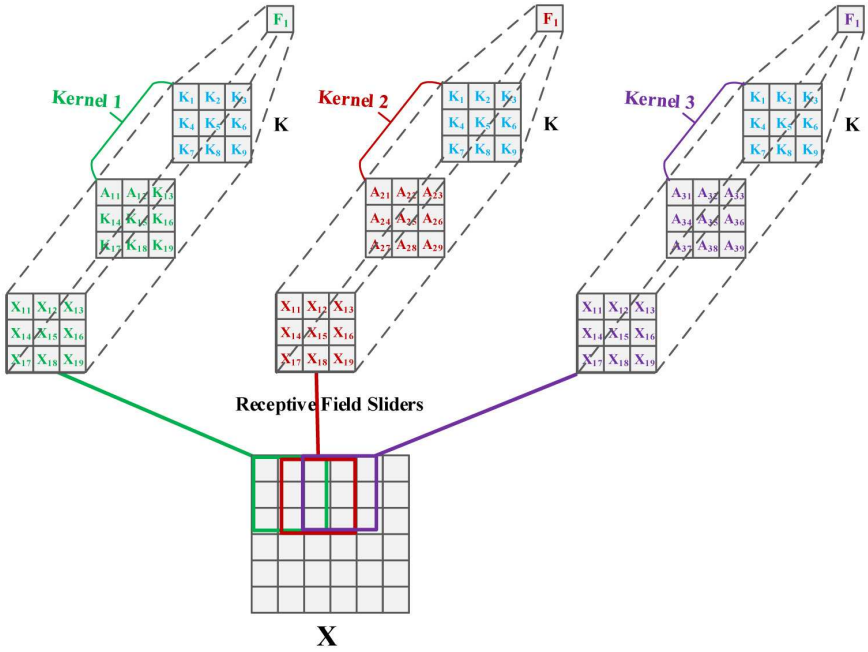

Fig. 4. It is obvious that there is an overlap of features in each receptive-field slider, which leads to the problem of sharing of attentional weights across sliders.

tures are obtained by multiplying the receptive-field slider, the attention weights and the 1×1 convolutional kernel parameters. In contrast to Fig. 1, if the product of the attention weights and the convolution kernel parameters is treated as a new convolution parameter, then the convolution parameter in each receptive field are different and not shared. However, when the spatial attention mechanism is inserted in front of the 3×3 convolutional operation, it will be limited. As mentioned above, if we take the value of $A_i \times K$ as a new convolution kernel parameter, the Equation (4) completely solves the problem of parameter sharing for large-scale convolutional kernels. As shown in the Fig. 4, it represents the process in which the input features are weighted by spatial attention and then operated by a 3×3 convolution. If the product of the attention weight corresponding to the receptive-field slider and the convolution kernel is taken as the convolution parameter, then the problem of convolution parameter sharing is solved. However, the most important point is that the convolutional kernel will share some of the features when it extracts features in each receptive field slider. In other words, there will be an overlap within each receptive-field slider, and when spatial attention is weighted, the size of the attention map is the same as the size of the input features, so the attention weights are shared in the sense field slider.

After careful analysis will find that $A_{12} = A_{21}, A_{13} = A_{22}, A_{15} = A_{24}, ...$ In this case, the weights of the spatial attention map are shared across each sliding window. As a result, the spatial attention mechanism cannot effectively address the problem of parameter sharing for large-scale convolutional kernels, such as the 3×3 convolution, because it does not consider the spatial features of the entire receptive field. Consequently, the effectiveness of the spatial attention mechanism is limited.

3.4 Innovating Spatial Attention and Standard Convolutional Operation

RFA is proposed to address the limitations of the existing spatial attention mechanism and provides an innovative solution to spatial processing. Drawing inspiration from RFA, a series of spatial attention mechanisms have been developed that can further enhance the performance of convolutional neural networks. RFA can be considered as a lightweight plug-and-play module, and the convolutional operation (RFAConv) designed by RFA can replace the standard convolution to improve the performance of convolutional neural networks.The spatial attention mechanism can solve the problem of convolutional parameter sharing by focusing on the receptive-field spatial feature, which is the direction for upgrading the existing spatial attention. The performance can be improved by combining the upgraded spatial features with the convolutional operation. So, we predict that the combination of spatial attention mechanisms and standard convolution operations will continue to evolve and lead to new breakthroughs in the future.

Receptive-Field Spatial Feature: In order to better understand the concept of receptive-field spatial feature, We will provide the relevant definition. The receptive-field spatial feature is specifically designed for convolutional kernels and is dynamically generated based on the kernel size. As shown in Fig. 5, the 3×3 convolutional kernel is used as an example.

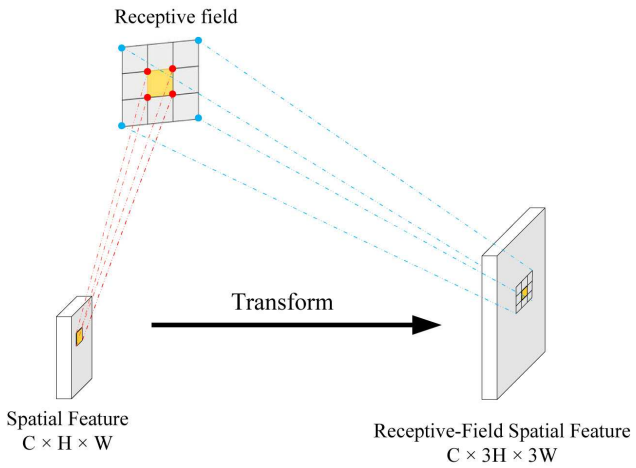

Fig. 5. The receptive-field spatial features are obtained by transforming the spatial features.

In Figure 5, the "Spatial Feature" refers to the original feature map. The "Receptive-Field Spatial Feature" is the feature map transformed by spatial features, which is composed of non-overlapping sliding windows. Each 3×3 size window in the receptive-field spatial feature represents a receptive-field slider.

Rceptive-Filed Attention Convolution (RFAConv): Regarding the receptive-field spatial feature, we propose receptive-field attention (RFA). This approach not only emphasizes the significance of different features within the receptive field slider, but also prioritizes the receptive-field spatial feature. Through this method, the problem of convolution kernel parameter sharing is completely solved. The receptive-field spatial features are dynamically generated depending on the size of the convolution kernel, therefore, The RFA is a fixed combination of convolution and cannot be separated from the help of convolution operations, which simultaneously rely on RFA to improve performance, so we propose receptive-field attention convolution (RFAConv). The overall structure of RFAConv with a 3×3 size convolutional kernel is shown in Fig. 7. As one of the popular frameworks in the field of deep learning, Pytorch provides the Unfold method to extract receptive-field spatial features. The detailed structure is shown in Fig. 6, which extracts 3×3 receptive-field spatial features. Let input X ∈ $R^{C\times H\times W}$, after the Unfold method, its dimension becomes 9C×H×W. Where C, H, and W represent the number of channels, height and width of input. Although Unfold is able to extract receptive-field spatial features by parameterless way, it is slow. Therefore, in RFAConv, we utilize a fast method to extract receptive-field spatial features, i.e., Group Conv. As mentioned in the previous section, each 3×3 size window in the receptive-field spatial feature represents a receptive-field slider when the 3×3 convolution kernel is used to extract features. However, after using fast Group Conv to extract the receptive-field features, original features are mapped into new features. This method is fast and more efficient than the original Unfold method. As shown in Table 1, experiments based on the YOLOv5n and VisDrone dataset [40] demonstrate it. It can be seen that RFAConv based on GroupConv obtains good performance while training 300 epochs requires less training time than the Unfold method. Morefore, we need to explain that the Unfold method is parameterless, while in the Table 1, it can be seen that the number of parameters required for the GroupConv-based method is the same as for the Unfold method. Because we use a lightweight approach to interacting information in receptive-field.

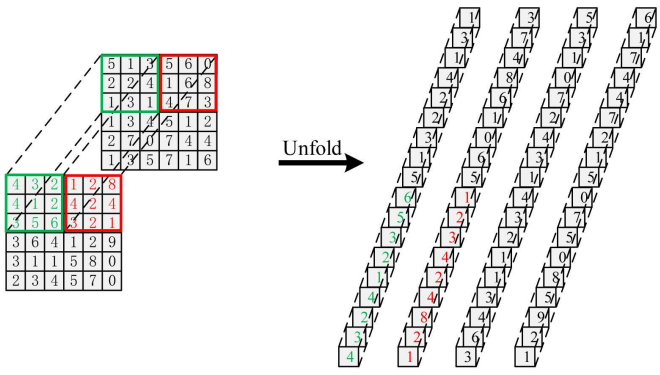

Fig. 6. In the figure, it shows in detail an example of extracting 3×3 receptive-field spatial features by the Unfold method. For the convenience of display, we set the step size to 3.

Recent research has shown that interacting information can enhance network performance, as demonstrated in [41, 42, 43]. Similarly, for RFAConv, interacting receptive-field feature information to learn the attention map can enhance network performance. However, interacting with each receptive-field feature can result in additional computational overhead,so to minimize computational overhead and number of parameters, the AvgPool is utilized to aggregate the global information of each receptive-field feature. Then, a 1×1 group convolutional operation is used to interact information. Finally, we use softmax to emphasize the significance of each feature within the receptive-field feature. In general, the calculation of RFA can be expressed as:

$$\begin{aligned} F &= Softmax(g^{1\times1}(AvgPool(X))) \times ReLU(Norm(g^{k\times k}(X))) \\ &= A_{rf} \times F_{rf} \end{aligned} \tag{5}$$

Here, $g^{i\times i}$ represents a grouping convolution of size $i \times i$, k represents the size of the convolution kernel, Norm stands for normalization, X represents the input feature maps, F is obtained by multiplying the attention map A_{rf} with the transformed receptive-field spatial feature F_{rf}. Unlike CBAM and CA, RFA is capable of generating attention maps for each receptive-field feature. The performance of convolutional neural networks is limited by the standard convolutional operations due to the fact that the convolution operation relies on shared parameters and is not sensitive to differences in information brought about by positional variations. However, RFAConv can completely address this issue by emphasizing the significance of different features within the receptive field slider and prioritizing the receptive-field spatial feature.

The feature map obtained through RFA is the receptive-field spatial feature, which does not overlap after "Adjust Shape". Therefore, the learned attention map aggregate the feature information of each receptive field slider. In other words, the attention map is no longer shared within each receptive-field slider. This completely compensates for the shortcomings of the existing CA and CBAM attention mechanisms. RFA provides significant benefits to the standard convolution kernel. However, after adjusting shape, the features are k times in height and width, requiring a k × k convolution operation with a stride = k to extract feature information. The convolution operation RFAConv designed by RFA

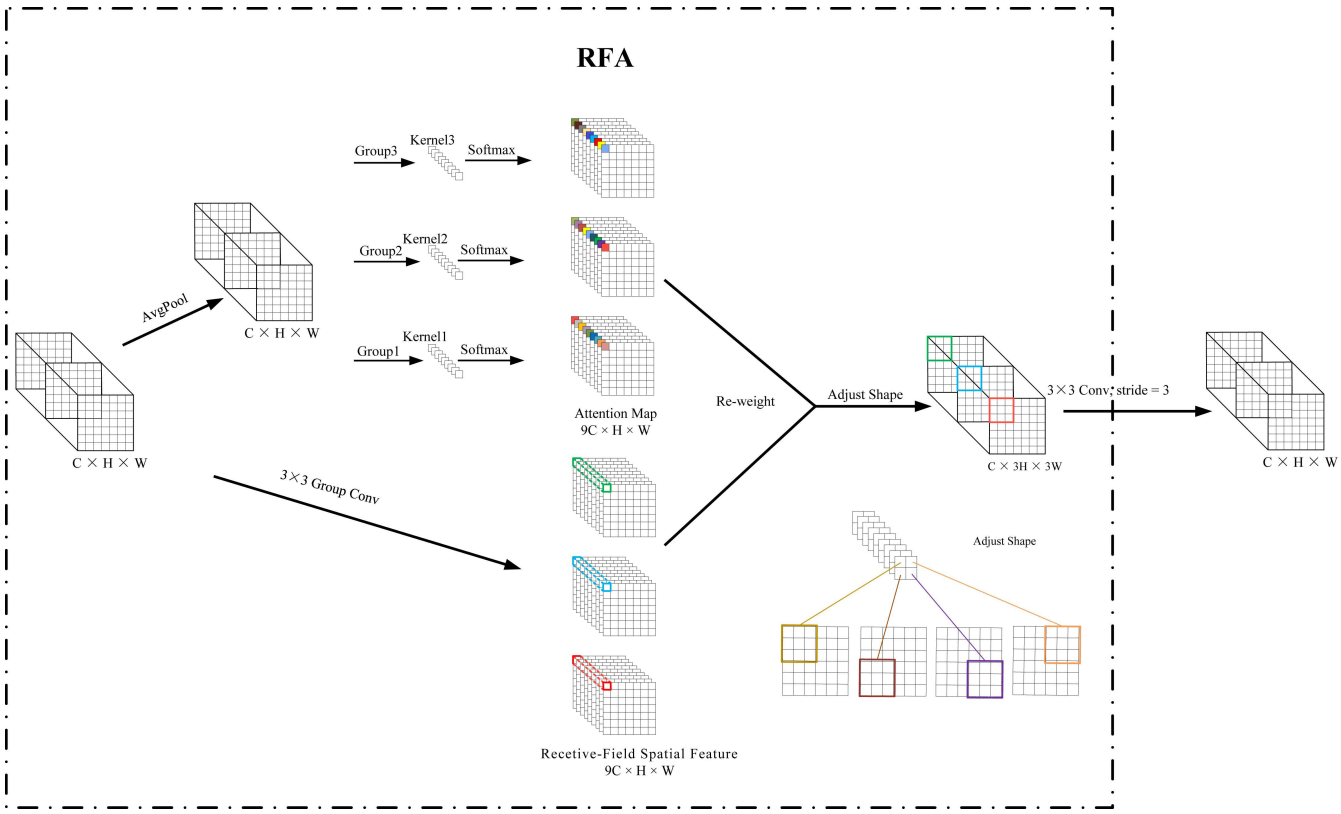

Fig. 7. The detailed structure of RFAConv, which dynamically determines the importance of each feature in the receptive-field and solves the problem of parameters sharing.

Methods	mAP50(%)	mAP(%)	FLOPS(G)	Param(M)	Training Time (Hours)
YOLOv5n	26.43	13.66	4.3	1.78	6.81
YOLOv5n + RFAConv (Unfold)	27.43	14.22	4.6	1.85	10.42
YOLOv5n + RFAConv (GroupConv)	27.58	14.36	4.7	1.85	7.37

Table 1. Object detection experiments based on YOLOv5n and VisDrone datasets to illustrate the advantages of RFAConv built on GroupConv.

brings good gains for convolution, and it innovates the standard convolution.

Furthermore, We assert that existing spatial attention mechanisms, such as CBAM [17] and CA [18], should prioritize receptive field spatial features to improve network performance. As is well-known, the network model based on the self-attention mechanism [44, 45, 46] has achieved great success, because it solves the problem of convolution parameter sharing and models long-distance information. However, the self-attention mechanism also introduces significant computational overhead and complexity to the model. We argue that directing the attention of existing spatial attention mechanisms to the receptive-field spatial feature can solve the problems of parameter sharing and modeling of long-range information in a way similar to self-attention. This approach requires significantly fewer parameters and computational resources than self-attention. The answers are as follows:

(1) The combination of the spatial attention mechanism, which focuses on the receptive-field spatial feature, with convolution eliminates the problem of convolution parameter sharing. (2) The current spatial attention mechanism already considers long-distance information and can obtain global information through global average pooling or global maximum pooling, which explicitly takes into account long-range information.

Therefore, we design new CBAM and CA models called RFCBAM and RFCA, which focus on the receptive-field spatial feature. Similar to RFA, the final convolution operation of $k \times k$ with stride = k is used to extract the feature information. The specific structure of these two new convolution methods, as shown in Fig. 8, we call these two new convolution operations RFCBAMConv and RFCAConv. Comparing the original CBAM, We use SE attention to replace CAM in RFCBAM. Because this can reduce computational overhead. Moreover in RFCBAM, channel and spatial attention are not performed in separate steps. Instead, they are weighted simultaneously, allowing the attention map obtained for each channel to be different.

4 Experiments and Discussions

To verify the effectiveness of our method, we perform classification, object detection and semantic segmentation experiments. The equipment for all experiments are based on RTX3090. In the classification experiments, we use four RTX3090 to train the model in parallel.

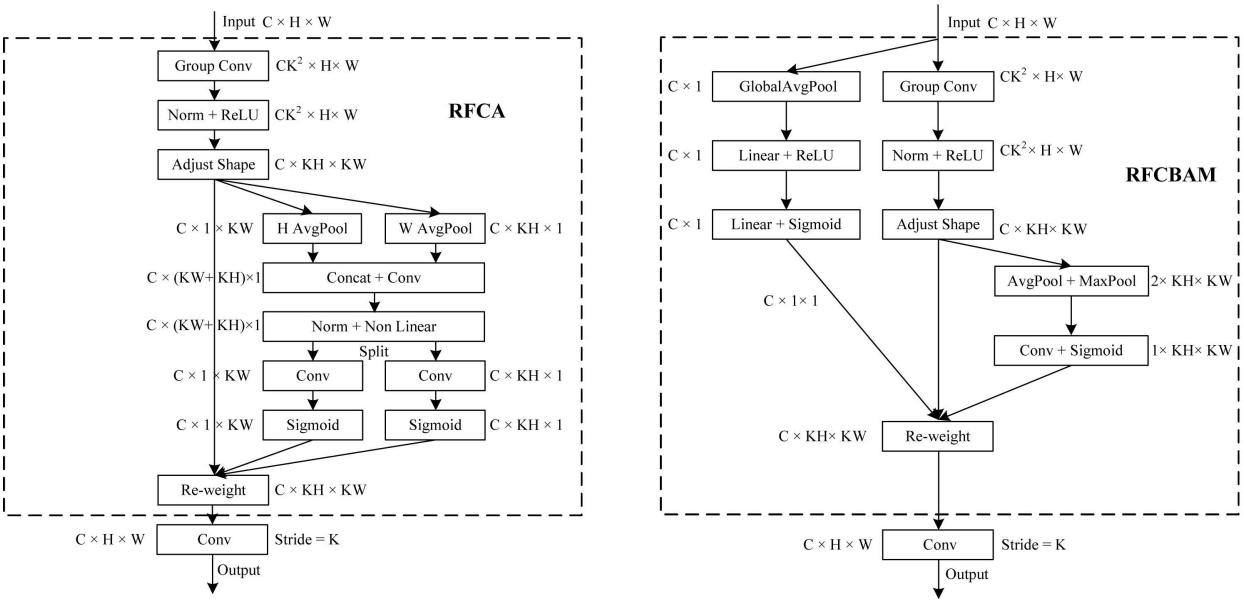

Fig. 8. Detailed structure of RFCAConv and RFCBAMConv, which focus on receptive-field spatial features. Comparing the original CBAM, We use SE attention to replace CAM in RFCBAM. Because, this can reduce computational overhead.

Layer Name	Output Size	Resnet18	Resnet34
Conv1	112 × 112		
Layer1	56 × 56	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 2$	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 3$
Layer2	28 × 28	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 2$	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 4$
Layer3	14 × 14	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 2$	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 6$
Layer4	7 × 7	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 2$	$\begin{bmatrix} NewConv\ 3\times3 \\ Conv\ \ 3\times3 \end{bmatrix}\times 3$
	1 × 1	AvgPool 1000-d	

Table 2. The Resnet18 and Resnet34 are construct by the new convolution operation.

4.1 Classification experiments on ImageNet-1k

We perform experiments to validate our method on the ImageNet-1k, which contains a number of 1281167 training sets and 50000 validation sets. Similar to RFAConv, we construct CBAMConv and CAConv by combining the CBAM and CA, respectively, with an additional 3 × 3 convolution layer at the end of the CBAM and CA modules. We also compare the CAMConv, which is constructed using the channel attention mechanism CAM [17]. we evaluate in ResNet18, ResNet34. Specifically, RFAConv, CBAMConv, CAConv, and CAMConv are used to replace (r) the first convolutional layer of BasicBlock in ResNet18 and ResNet34, respectively. In general, the new convolution is structured as shown in Table 2.

The NewConv in Table 2 represents the convolution mode constructed by the attention mechanism. For CAConv, CBAMConv, and CAMConv, it can be considered that CA, CBAM, and CAM are added before the first layer of convolution in BasicBlock. In the image classification experiments, we train 100 epochs for each model with batch-size of 128. The learning rate start from 0.1 and decrease every 30 epochs, 0.1 times each time. In experiments, we follow most of the previous work and report the accuracy for TOP1 and TOP5, respectively. Table 3 shows the results produced by different networks on the ImageNet-1K validation set. It is clear that replacing the 3×3 convolutional operation with RFAConv significantly improves the recognition results. Compared to the baseline models RestNet18 and ResNet34 , the network constructed by RFAConv achieves the best recognition results at the cost of only a small increase in parameters and computational overhead. Such as ResNet18 constructed based on RFAConv only adds 0.16 M parameters and 0.09 G computational overhead over the original model, and increases the accuracy by 1.64% and 1.24% on TOP1 and TOP5, respectively.

Moreover, as we mentioned earlier, spatial attention can be enhanced again by focusing on receptive-field spatial features. Therefore, we design RFCBAMConv and RFCAConv, which are improvements of CBAM and CA. In order to verify their advantages, we conduct experiments based on ResNet18 and report the relevant data in Table 4. It is obvious that RFCBAMConv and RFCAConv achieve better recognition accuracy, compared with CBAMConv and CAConv in Table 3. Most importantly, they also significantly improve performance at the cost of only a small increase in parameters and computational overhead. This is a strong demonstration that spatial attention can be improved by placing attention into the receptive-field spatial features. It fully demonstrates

Models	FLOPS(G)	Param(M)	Top1(%)	Top5(%)
Resnet18	1.82	11.69	69.59	89.05
+ CAMConv(r)	1.83	11.75	70.76	89.74
+ CBAMConv(r)	1.83	11.75	69.38	89.12
+ CAConv(r)	1.83	11.74	70.58	89.59
+ RFAConv(r)	1.91	11.85	**71.23**	**90.29**
Resnet34	3.68	21.80	73.33	91.37
+ CAMConv(r)	3.68	21.93	74.03	91.69
+ CBAMConv(r)	3.68	21.93	72.95	91.26
+ CAConv(r)	3.68	21.91	73.76	91.68
+ RFAConv(r)	3.84	22.16	**74.25**	**92.03**

Table 3. Classification results on ImageNet-1K using the Resnet18 and Resnet34. The different convolutional operation constructed by the attention mechanism is compared.

that spatial attention can be improved by placing attention into the receptive-field spatial features.

Models	FLOPS(G)	Param(M)	Top1(%)	Top5(%)
Resnet18	1.82	11.69	69.59	89.05
+ RFCBAMConv(r)	1.90	11.88	**72.15**	**90.71**
+ RFCAConv(r)	1.92	11.89	72.01	90.64

Table 4. RFCBAMConv and RFCAConv improve the performance of CBAMConv and CAConv. The table shows that the classification accuracy is significantly improved on ImageNet-1k.

All classification experiments clearly demonstrate the significant advantages of our methods, because RFAConv, RFCBAMConv, and RFCAConv completely address the problem of convolution kernel parameter sharing. Moreover, it is worth noting that RFCBAMConv and RFCAConv outperform RFAConv, because they not only solve the problem of convolution kernel parameter sharing but also consider long-distance information through global pooling.

Moreover, to provide a more intuitive analysis, as with the most work, we use the Grad-CAM [47] algorithm for visualization. Grad-CAM highlights the regions of interest of different networks for a particular class of objects. To some extent, it is possible to see how the networks utilize the features. We randomly selected some images in the validation set of ImageNet-1K and visualized the results of the networks constructed with different attention convolutions based on ResNet18 separately. As shown in the Fig. 9, compared with other attention convolutions, our RFAConv can help the network to better recognize and highlight the key regions of objects.

We also use Grad-CAM to visualize ResNet18 constructed by CBAMConv, RFCBAMConv, CAConv, and RFCAConv. RFCBAM is obtained by improving the attention of CBAM by putting it into the recepitve-field spatial feature. Similarly, RFCA is obtained by CA after the same method. As shown in Fig. 10, the improved RFCA and RFCBAM can help the network to better recognize and highlight the key regions of objects after combining with the convolution operation.

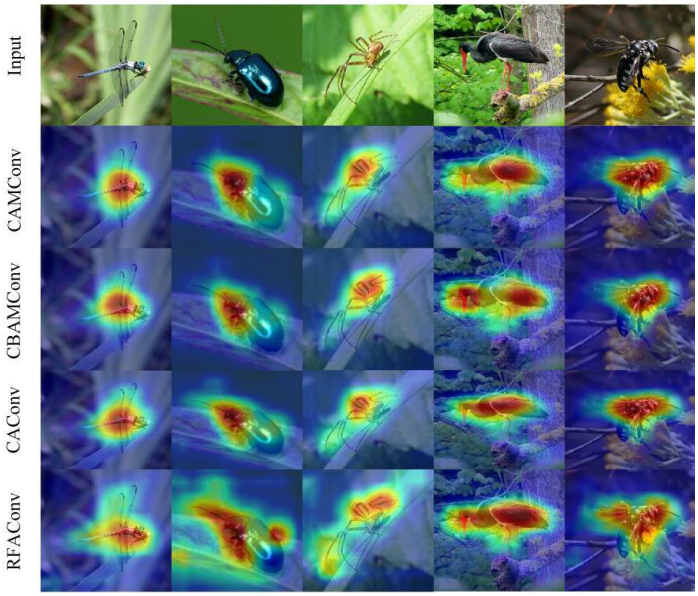

Fig. 9. Each network is built on ResNet18 based on attention convolution, and the construction process is shown in Table 2. We use Grad-CAM as our visualization tool to visualize networks without the last layer of classifiers. Compared to other attention convolution methods, our RFAConv can help the network to better recognize and highlight the key regions of objects

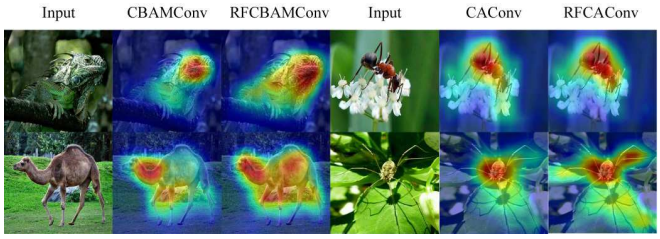

Fig. 10. We put the attention of CBAM and CA into the receptive-field spatial features and improve them to obtain RFCBAM and RFCA. Then RFCBAMConv and RFCAConv are constructed by the same method as RFA. As in Fig. 10, we visualize the different networks separately. Obviously, compared to CBAMConv and CAConv, the improved obtained RFCBAMConv and CAConv can help the network to better recognize and highlight the key regions of objects.

4.2 Object detection experiments on COCO2017

We conduct object detection experiments on COCO2017 to re-evaluate our methods. COCO2017 contains 118287 training sets and 5000 verification sets. We select YOLOv5n, YOLOv7-tiny, and YOLOv8n models to perform a series of experiments. All parameters except for epoch and batch-size are set to the default values. We train each model for 300 epochs with a batch-size of 32. To be similar in classification, we replace some convolution operation in the baseline model with novel convolution operations constructed using attention mechanisms. Specifically, we replace all 3×3 convolution operations in the yaml files for YOLOv5 and YOLOv8 using attention convolution. And for YOLOv7, we replace the first 3×3 convolution operation in all ELAN [34] in the backbone. Following the previous work, we report AP_{50}, AP_{75}, AP, AP_S, AP_M and AP_L separately. Moreover, in order to better display the performance of different networks, we chose the training process of YOLOv5n for visualization. We visualize how the AP50 changes with the number of iterations.

Models	FLOPS(G)	Param(M)	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	Time(ms)
YOLOv5n	4.5	1.8	45.6	28.9	27.5	13.5	31.5	35.9	4.4
+ CAMConv(r)	4.5	1.8	45.6	28.3	27.4	13.8	31.4	35.8	5.2
+ CBAMConv(r)	4.5	1.8	45.5	28.6	27.6	13.6	31.2	36.6	5.4
+ CAConv(r)	4.5	1.8	46.2	29.2	28.1	14.3	32	36.6	4.8
+ RFAConv(r)	4.7	1.9	**47.3**	**30.6**	**29**	**14.8**	**33.4**	**37.4**	5.3
YOLOv7-tiny	13.7	6.2	53.8	38.3	35.9	19.9	39.4	48.8	6.8
+ RFAConv(r)	14.1	6.3	**55.1**	**40.1**	**37.1**	**20.9**	**41.1**	**50**	8.4
YOLOv8n	8.7	3.1	51.9	39.7	36.4	18.4	40.1	52	4.2
+ CAMConv(r)	8.8	3.1	51.6	39	36.2	18	39.9	51.2	4.5
+ CBAMConv(r)	8.8	3.1	51.5	39.6	36.3	18.3	40.1	51.5	4.6
+ CAConv(r)	8.8	3.1	52.1	39.9	36.7	17.8	40.3	51.6	4.3
+ RFAConv(r)	9.0	3.2	53.4	41.1	37.7	18.9	41.8	52.7	4.5
+ RFCAConv(r)	9.1	3.2	**53.9**	**41.7**	**38.2**	**19.7**	**42.3**	**53.5**	4.7

Table 5. Object detection AP_{50}, AP_{75}, AP, AP_S, AP_M, and AP_L on the COCO2017 validation sets. We adopt the YOLOv5n, YOLOv7-tiny, and YOLOv8n detection framework and replace the original convolution with the novel convolutioanl operation constructed by attention mechanism.

The experimental results are shown in Table 5 and Fig. 11. When RFAConv is used to replace some convolution, the network achieved significantly improved detection results with only a small increase in the number of parameters and computational overhead. Compared to other attention, RFA still brings considerable benefits to the detection network. In some experiments, we once again verify the effectiveness of RFCA, which exhibits better performance of convolutional operation compared to the original CA. Time represents the total time spent processing an image during validation. It can be clearly seen that the model constructed with the novel convolutioanl operation has an increase in time for processing a image. Therefore, if real-time is pursued, the number of replacement convolutions should not be too many.

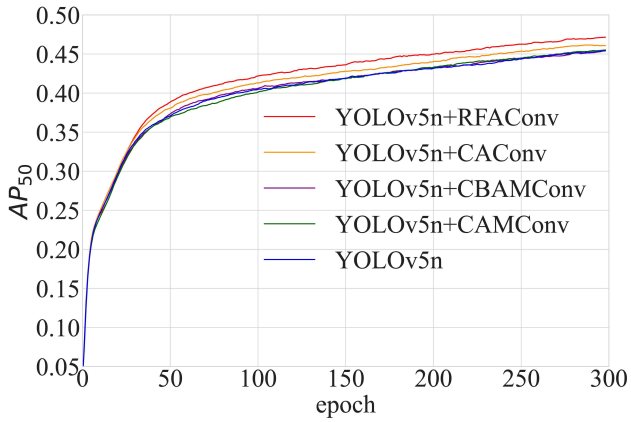

Fig. 11. The AP_{50} change during training for the different YOLOv5n constructed by attention convolution.

4.3 Object detection experiments on VOC7+12

In order to validate our method again, we select the VOC7+12 dataset for experiments. VOC7+12 is a mixture of VOC2007 and VOC2012, with a total of 16551 training sets and 4952 verification sets. Similar to the experiments on COCO2017, we conduct experiments on advanced detection models such as YOLOv5n, YOLOv5s, YOLOv7-tiny, and YOLOv8n. All hyperparameter settings and network structure are the same as in the previous section.

Following most of the work, we also report mAP. As shown in Table 6. As with the previously obtained conclusions, in all experiments, after we replace some of the convolution operations in the network using RFAConv, the network gained significant improvement by adding only a small number of parameters and computational overhead. Meanwhile, RFA obtains an outstanding performance compared to other attentions. Moreover, in some experiments, in some experiments, we similarly experiment with networks constructed by RFCBAMConv and RFCAConv. The results again validate their advantages. Also compared with CBAMConv and CAConv, they still obtained better results.

4.4 Semantic segmentation experiments on VOC2012

In order to validate the advantages of our method again, we conduct semantic segmentation experiments on the VOC2012 dataset, selecting DeepLabplusV3 [48] and the backbone network Resnet18 to conduct related experiments. The pretraining weights of each backbone network are obtained in the ImageNet-1k experiment. We report the results for outputs at two different step sizes, 8 and 16, respectively. In the experiment, we found that the semantic segmentation network constructed by RFAConv achieved better results than the original model, but compared to CAConv, CAMConv, the performance of RFAConv is not good. After thinking, we assert that RFAConv lacks consideration for long-distance information, while semantic segmentation tasks rely on long-distance information. CAConv, CAMConv and CBAMConv capture long range information by global averaging pooling to obtain global information. Although CBAMConv produce poor results for semantic segmentation, the improved RFCBAM obtain the good performance. This again demonstrates that spatial attention can again improve network per-

Models	FLOPS(G)	Param(M)	mAP(%)	Time(ms)
YOLOv5n	4.2	1.7	41.5	2.7
+ CAMConv(r)	4.2	1.7	41.4	2.9
+ CBAMConv(r)	4.3	1.7	41.9	3
+ CAConv(r)	4.3	1.7	42.4	3
+ RFAConv(r)	4.5	1.8	**43.3**	3
YOLOv5s	15.9	7.1	48.9	3
+ CAMConv(r)	16	7.1	48.5	3.5
+ CBAMConv(r)	16	7.1	49	3.7
+ CAConv(r)	16.1	7.1	49.6	3.1
+ RFAConv(r)	16.4	7.2	50	5.1
+ RFCBAMConv(r)	16.4	7.2	50.1	3.9
+ RFCAConv(r)	16.6	7.2	**51**	4.4
YOLOv7-tiny	13.2	6.1	50.2	5
+ CAMConv(r)	13.2	6.1	50.3	5.4
+ CBAMConv(r)	13.2	6.1	50.1	5.4
+ CAConv(r)	13.2	6.1	50.5	5.4
+ RFAConv(r)	13.6	6.1	**50.6**	7.5
YOLOv8n	8.1	3	53.5	3
+ CAMConv(r)	8.1	3	52.8	3.1
+ CBAMConv(r)	8.2	3	53.3	3.1
+ CAConv(r)	8.2	3	53.8	2.9
+ RFAConv(r)	8.4	3.1	**54**	3.2

Table 6. Object detection mAP50 and mAP on the VOC7+12 validation set.

formance through our approach, which simply requires making spatial attention attend to the receptive-field spatial feature.

4.5 Discussions

As all the experiments have shown, RFAConv serves as an alternative to standard convolution, bringing significant improvements to classification, target detection, and semantic segmentation visual tasks at the cost of a small increase in parameters and computational effort. As shown in the classification visualization in Fig. 9 and Fig. 10, networks built based on RFAConv are able to better focus on important information and features. This is due to the fact that RFA takes into account the receptive-field spatial features and is able to highlight the importance of each feature within the receptive-field, while combining it with convolution to turn it into a non-parametric shared convolution operation. Also as mentioned earlier, increasing the focus of CA and CBAM to the receptive-field spatial feature can again improve performance. Thus RFCBAM,RFCA was designed by us to significantly improve the network performance. What is clear is that RFAConv is advantageous but not outstanding in semantic segmentation tasks. We think it is that the design of RFA does not take long range information into account. As Hou et al.[18] argued, semantic segmentation relies on long range information. In contrast to RFA, RFCBAM and RFCA take into account the receptive-field spatial features and consider long range information through maximum pooling and average pooling. So they obtain outstanding performance in every visual task. By carefully analyzing RFAConv, RFCBAMConv and RFCAConv, our task can put some existing spatial attention into the receptive field to enhance its performance.

Backbone	Stride	MIOU(%)
Resnet18	8	58.9
+ CAMConv(r)	8	60.9
+ CBAMConv(r)	8	59.3
+ CAConv(r)	8	62.1
+ RFAConv(r)	8	60.8
+ RFCBAMConv(r)	8	62.1
+ RFCAConv(r)	8	**63.9**
Resnet18	16	64.6
+ CAMConv(r)	16	65.5
+ CBAMConv(r)	16	63.6
+ CAConv(r)	16	66.6
+ RFAConv(r)	16	65.4
+ RFCBAMConv(r)	16	67.7
+ RFCAConv(r)	16	**68.0**

Table 7. Results of experiments comparing different the novel convolutional operation based on DeepLabPlusV3.

5 Conclusion

By analyzing the standard convolution and spatial attention, we conclude that spatial attention mechanisms address the problem of parameter sharing and have the advantage of considering long-distance information. However, the performance of the spatial attention mechanism is limited for the large-size convolution kernel. To address this issue, we propose a novel attention mechanism called RFA and devise a novel convolution operation, which further improves network performance. We also emphasize the importance of directing attention to the receptive-field spatial feature to enhance network performance. Through extensive experiments, we demonstrate the effectiveness and advanced nature of our approach. Going forward, we hope that more and more spatial attention mechanisms will adopt our proposed approach to further improve performance, and we also hope that the novel convolution method will be widely adopted to enhance network performance.

Bibliography

[1] M. Lin, Q. Chen, S. Yan, Network in network, arXiv preprint arXiv:1312.4400 (2013). 1

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

[3] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324. 1

[4] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90. 1

[5] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014). 1

[6] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258. 1

[7] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1314–1324. 1

[8] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, Ghostnet: More features from cheap operations, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1580–1589. 1

[9] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, et al., Resnest: Split-attention networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2736–2746. 1

[10] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986. 1

[11] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, J. Shlens, Stand-alone self-attention in vision models, Advances in neural information processing systems 32 (2019). 1

[12] G. Elsayed, P. Ramachandran, J. Shlens, S. Kornblith, Revisiting spatial invariance with low-rank local connectivity, in: International Conference on Machine Learning, PMLR, 2020, pp. 2868–2879. 1

[13] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang, Q. Chen, Involution: Inverting the inherence of convolution for visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12321–12330. 1

[14] J. Park, S. Woo, J.-Y. Lee, I. S. Kweon, Bam: Bottleneck attention module, arXiv preprint arXiv:1807.06514 (2018). 1

[15] X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 510–519. 1

[16] A. Luo, F. Yang, X. Li, S. Liu, Learning optical flow with kernel patch attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8906–8915. 1

[17] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19. 1, 2, 6, 7

[18] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13713–13722. 1, 2, 6, 10

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255. 1

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755. 1

[21] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International journal of computer vision 111 (2015) 98–136. 1

[22] M. Hassaballah, M. A. Kenk, K. Muhammad, S. Minaee, Vehicle detection and tracking in adverse weather using a deep learning framework, IEEE transactions on intelligent transportation systems 22 (7) (2020) 4230–4242. 1

[23] X. Zhu, S. Lyu, X. Wang, Q. Zhao, Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 2778–2788. 1

[24] Z. Ning, S. Zhong, Q. Feng, W. Chen, Y. Zhang, Smunet: saliency-guided morphology-aware u-net for breast lesion segmentation in ultrasound image, IEEE transactions on medical imaging 41 (2) (2021) 476–490. 1

[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778. 1

[26] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708. 1

[27] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 764–773. 1

[28] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 9308–9316. 2

[29] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al., Internimage: Exploring large-scale vision foundation models with deformable convolutions, arXiv preprint arXiv:2211.05778 (2022). 2

[30] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856. 2

[31] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, Shufflenet v2: Practical guidelines for efficient cnn architecture design, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131. 2

[32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788. 2

[33] J. Glenn, Yolov5 release v6.1, https://github.com/ultralytics/yolov5/releases/tag/v6.1 (2022). 2

[34] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, arXiv preprint arXiv:2207.02696 (2022). 2, 8

[35] J. Glenn, Ultralytics yolov8, https://github.com/ultralytics/ultralytics (2023). 2

[36] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141. 2

[37] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, Eca-net: Efficient channel attention for deep convolutional neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11534–11542. 2

[38] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 3146–3154. 2

[39] H. Zhang, K. Zu, J. Lu, Y. Zou, D. Meng, Epsanet: An efficient pyramid squeeze attention block on convolutional neural network, in: Proceedings of the Asian Conference on Computer Vision, 2022, pp. 1161–1177. 2

[40] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, H. Ling, Detection and tracking meet drones challenge, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (11) (2021) 7380–7399. 5

[41] I. Bello, B. Zoph, A. Vaswani, J. Shlens, Q. V. Le, Attention augmented convolutional networks, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 3286–3295. 5

[42] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, A. Vaswani, Bottleneck transformers for visual recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 16519–16529. 5

[43] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, J. Shlens, Scaling local self-attention for parameter efficient visual backbones, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12894–12904. 5

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017). 6

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020). 6

[46] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022. 6

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626. 8

[48] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818. 9