# EPSANet: An Efficient Pyramid Squeeze Attention Block on Convolutional Neural Network

Hu Zhang[1,2], Keke Zu[1,2], Jian Lu[1,2⋆], Yuru Zou[1,2], and Deyu Meng[3,4]

[1] Shenzhen Key Laboratory of Advanced Machine Learning and Applications, College of Mathematics and Statistics, Shenzhen University, Shenzhen, China
[2] National Center for Applied Mathematics Shenzhen (NCAMS), Shenzhen, China
[3] Xi'an Jiaotong University, Xi'an, China
[4] Macau University of Science and Technology, Macao, China
huzhang198@gmail.com, {kekezu,jianlu,yuruzou}@szu.edu.cn, dymeng@mail.xjtu.edu.cn

**Abstract.** Recently, it has been demonstrated that the performance of a deep convolutional neural network can be effectively improved by embedding an attention module into it. In this work, a novel lightweight and effective attention method named Pyramid Squeeze Attention (PSA) module is proposed. By replacing the 3x3 convolution with the PSA module in the bottleneck blocks of the ResNet, a novel representational block named Efficient Pyramid Squeeze Attention (EPSA) is obtained. The EPSA block can be easily added as a plug-and-play component into a well-established backbone network, and significant improvements on model performance can be achieved. Hence, a simple and efficient backbone architecture named EPSANet is developed in this work by stacking these ResNet-style EPSA blocks. Correspondingly, a stronger multi-scale representation ability can be offered by the proposed EPSANet for various computer vision tasks including but not limited to, image classification, object detection, instance segmentation, etc. Without bells and whistles, the performance of the proposed EPSANet outperforms most of the state-of-the-art channel attention methods. As compared to the SENet-50, the Top-1 accuracy is improved by 1.93% on ImageNet dataset, a larger margin of +2.7 box AP for object detection and an improvement of +1.7 mask AP for instance segmentation by using the Mask-RCNN on MS-COCO dataset are obtained.

**Keywords:** Computer Vision · Attention Module.

## 1 Introduction

Attention mechanisms are widely used in many computer vision areas such as image classification[1, 2], object detection[3], instance segmentation[4], semantic segmentation[5, 6], scene parsing and action localization[7]. Specifically, there

⋆ Corresponding author.

are two types of attention methods, which are channel attention and spatial attention. Recently, it has been demonstrated that significant performance improvements can be achieved by employing the channel attention[8, 9], spatial attention[10], or both of them[11]. The most commonly used method of channel attention is the Squeeze-and-Excitation (SE) module [12], which can significantly improve the performance with a considerably low cost. The drawback of the SENet is that it ignores the importance of spatial information. Therefore, the Bottleneck Attention Module(BAM) [10] and Convolutional Block Attention Module(CBAM) [11] are proposed to enrich the attention map by effectively combining the spatial and channel attention. However, there still exists two important and challenging problems. The first one is how to efficiently capture and exploit the spatial information of the feature map with different scales to enrich the feature space. The second one is that the channel or spatial attention can only effectively capture the local information but fail in establishing a long-range channel dependency. Correspondingly, many methods are proposed to address these two problems. The methods based on multi-scale feature representation and cross-channel information interaction, such as the PyConv [13], the Res2Net [14], and the HS-ResNet [15], are proposed. In the other hand, a long-range channel dependency can be established as shown in [6, 16, 17]. All the above mentioned methods, however, bring higher model complexity and thus the network suffers from heavy computational burden. Based on the above observations, in this work, a low-cost and high-performance novel module named Pyramid Squeeze Attention (PSA) is proposed. Firstly, the proposed PSA module uses the multi-scale pyramid convolution structure to process the input tensor at multiple scales. Secondly, the PSA module can effectively extract spatial information with different scales from each channel-wise feature map by squeezing the channel dimension of the input tensor. Third, a cross-dimension interaction can be built by extracting the channel-wise attention weight of the multi-scale feature maps. Finally, the softmax operation is employed to recalibrate the attention weight of the corresponding channels, and thus the interaction between the channels that are in different groups of the squeeze-concatenate module is established. Correspondingly, a novel block named Efficient Pyramid Squeeze Attention (EPSA) is obtained by replacing the 3x3 convolution with the PSA module in the bottleneck blocks of the ResNet. Furthermore, a novel backbone EPSANet is proposed by stacking these EPSA blocks as the ResNet style. The main contributions of this work are summarized as below:

- A novel Efficient Pyramid Squeeze Attention (EPSA) block is proposed, which can effectively extract multi-scale spatial information at a more granular level and develop a long-range channel dependency. The proposed EPSA block is very flexible and scalable and thus can be applied to a large variety of network architectures for numerous tasks of computer vision.
- A novel backbone architecture named EPSANet is proposed, which can learn richer multi-scale feature representation and adaptively re-calibrate the cross-dimension channel-wise attention weight.

– Extensive experiments demonstrated that promising results can be achieved by the proposed EPSANet across image classification, object detection and instance segmentation on both ImageNet and COCO datasets.

## 2 Related Work

**Attention mechanism.** The attention mechanism is used to strength the allocation of the most informative feature expressions while suppressing the less useful ones, and thus makes the model attending to important regions within a context adaptively. The Squeeze-and-Excitation (SE) attention in [12] can capture channel correlations by selectively modulating the scale of channel. The CBAM in [11] can enrich the attention map by adding max pooled features for the channel attention with large-size kernels. Motivated by the CBAM, the GSoP in [18] proposed a second-order pooling method to extract richer feature aggregation. More recently, the Non-Local block [17] is proposed to build a dense spatial feature map and capture the long-range dependency via non-local operations. Based on the Non-Local block, the Double Attention Network($A^2$Net) [19] introduces a novel relation function to embed the attention with spatial information into the feature map. Sequently, the SKNet in [20] introduces a dynamic selection attention mechanism that allows each neuron to adaptively adjust its receptive field size based on multiple scales of input feature map. The ResNeSt [21] proposes a similar Split-Attention block that enables attention across groups of the input feature map. The Fcanet [8] proposes a novel multi-spectral channel attention that realizes the pre-processing of channel attention mechanism in the frequency domain. The GCNet [1] introduces a simple spatial attention module and thus a long-range channel dependency is developed. The ECANet [9] employs the one-dimensional convolution layer to reduce the redundancy of fully connected layers. The DANet [16] adaptively integrates local features with their global dependencies by summing these two attention modules from different branches. The above mentioned methods either focus on the design of more sophisticated attention modules that inevitably bring a greater computational cost, or they cannot establish a long-range channel dependency. Thus, in order to further improve the efficiency and reduce the model complexity, a novel attention module named PSA is proposed, which aims at learning attention weight with low model complexity and to effectively integrate local and global attention for establishing the long-range channel dependency.

**Multi-scale Feature Representations.** The ability of the multi-scale feature representation is essential for various vision tasks such as, instance segmentation [4], face anaxlysis [22], object detection [23], salient object detection [24], and semantic segmentation [5]. It is critically important to design a good operator that can extract multi-scale feature more efficiently for visual recognition tasks. By embedding a operator for multi-scale feature extraction into a convolution neural network(CNN), a more effective feature representation ability can be obtained. In the other hand, CNNs can naturally learn coarse-to-fine multi-scale features through a stack of convolutional operators. Thus, to design a better

convolutional operator is the key for improving the multi-scale representations of CNNs.

## 3 Method

### 3.1 Revisting Channel Attention

**Channel attention** The channel attention mechanism allows the network to selectively weight the importance of each channel and thus generates more informative outputs. Let $X \in \mathbb{R}^{C\times H\times W}$ denotes the input feature map, where the quantity $H$, $W$, $C$ represent its height, width, number of input channels respectively. A SE block consists of two parts: squeeze and excitation, which is respectively designed for encoding the global information and adaptively recalibrating the channel-wise relationship. Generally, the channel-wise statistics can be generated by using a global average pooling, which is used to embed the global spatial information into a channel descriptor. The global average pooling operator can be calculated by the following equation

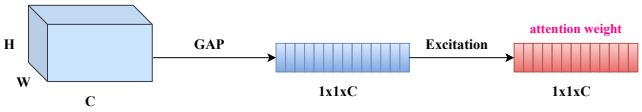


**Fig. 1.** SEWeight module.

$$g_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j) \tag{1}$$

The attention weight of the $c$-th channel in the SE block can be written as

$$w_c = \sigma(W_1 \delta(W_0(g_c))) \tag{2}$$

where the symbol $\delta$ represents the Recitified Linear Unit (ReLU) operation as in [25], $W_0 \in \mathbb{R}^{C\times \frac{C}{r}}$ and $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ represent the fully-connected (FC) layers. With two fully-connected layers, the linear information among channels can be combined more efficiently, and it is helpful for the interaction of the information of high and low channel dimensions. The symbol $\sigma$ represents the excitation function, and a sigmoid function is usually used in practice. By using the excitation function, we can assign weights to channels after the channel interaction and thus the information can be extracted more efficiently. The above introduced process of generating channel attention weights is named SEWeight module in [12], the diagram of the SEWeight module is shown by Figure 1.

### 3.2 PSA Module

The motivation of this work is to build a more efficient and effective channel attention mechanism. Therefore, a novel pyramid squeeze attention (PSA) module is proposed. As illustrated by Figure 3, the PSA module is mainly implemented in four steps. First, the multi-scale feature map on channel-wise is obtained by implementing the proposed squeeze pyramid concat (SPC) module. Second, the channel-wise attention vector are obtained by using the SEWeight module to extract the attention of the feature map with different scales. Third, re-calibrated weight of multi-scale channel is obtained by using the Softmax to re-calibrate the channel-wise attention vector. Fourth, the operation of an element-wise product is applied to the re-calibrated weight and the corresponding feature map. Finally, a refined feature map which is richer in multi-scale feature information can be obtained as the output. As illustrated by Figure 2, the essential operator for implementing the multi-scale feature extraction in the proposed PSA is the SPC, we extract the spatial information of the input feature map in a multi-branch way, the input channel dimension of each branch is $C$. By doing this, we can obtain more abundant positional information of the input tensor and process it at multiple scales in a parallel way. Thus a feature map that contains a single type of kernel can be obtained. Correspondingly, the different spatial resolutions and depths can be generated by using multi-scale convolutional kernels in a pyramid structure. And the spatial information with different scales on each channel-wise feature map can be effectively extracted by squeezing the channel dimension of the input tensor. Finally, each featur map with different scales $F_i$ has the common channel dimension $C^{'} = \frac{C}{S}$ and $i = 0, 1, \cdots, S-1$. Note that $C$ should be divisible by $S$. For each branch, it learns the multi-scale spatial information independently and establishs a cross-channel interaction in a local manner. However, a huge improvement in the amount of parameters will be resulted with the increase of kernel sizes. In order to process the input tensor at different kernel scales without increasing the computational cost, a method of group convolution is introduced and applied to the convolutional kernels. Further, we design a novel criterion for choosing the group size without increasing the number of parameters. The relationship between the multi-scale kernel size and the group size can be written as

$$G = \begin{cases} 2^{\frac{K-1}{2}} & K > 3 \\ 1 & K = 3 \end{cases} \tag{3}$$

where the quantity $K$ is the kernel size, $G$ is the group size. Finally, the multi-scale feature map generation function is given by

$$F_i = \mathrm{Conv}(k_i \times k_i, G_i)(X) \quad i = 0, 1, 2 \cdots S-1 \tag{4}$$

where the $i$-th kernel size $k_i = 2 \times (i+1) + 1$, the $i$-th group size $G_i$ and $F_i \in R^{C^{'} \times H \times W}$ denotes the feature map with different scales. The whole multi-scale pre-processed feature map can be obtained by a concatenation way as

$$F = \mathrm{Cat}([F_0, F_1, \cdots, F_{S-1}]) \tag{5}$$

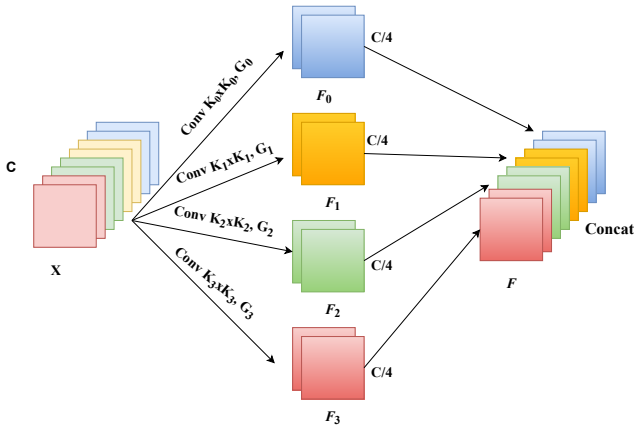


**Fig. 2.** A detailed illustration of the proposed Squeeze Pyramid Concat (SPC) module with $S$=4, where 'Squeeze' means to equally squeeze in the channel dimension, $K$ is the kernel size, $G$ is the group size and 'Concat' means to concatenate features in the channel dimension.

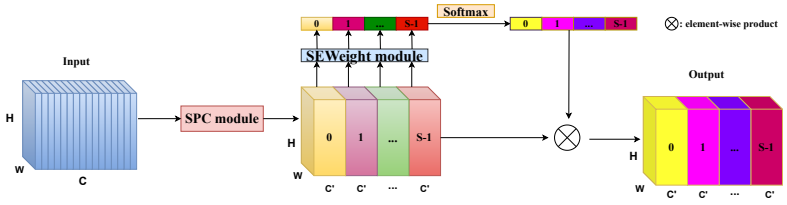


**Fig. 3.** The structure of the proposed Pyramid Squeeze Attention (PSA) module.

where $F \in R^{C\times H\times W}$ is the obtained multi-scale feature map. By extracting the channel attention weight information from the multi-scale pre-processed feature map, the attention weight vectors with different scales are obtained. Mathematically, the vector of attention weight can be represented as

$$Z_i = \mathrm{SEWeight}(F_i), \quad i = 0, 1, 2 \cdots S-1 \tag{6}$$

where $Z_i \in R^{C^{'}\times 1\times 1}$ is the attention weight. The SEWeight module is used to obtain the attention weight from the input feature map with different scales. By doing this, our PSA module can fuse context information in different scales and produce a better pixel-level attention for high-level feature maps. Further, in order to realize the interaction of attention information and fuse the cross-dimensions vector without destroying the orginal channel attention vector. And

thus the whole multi-scale channel attention vector is obtained in a concatenation way as

$$Z = Z_0 \oplus Z_1 \oplus \cdots \oplus Z_{S-1} \tag{7}$$

where $\oplus$ is the concat operator, $Z_i$ is the attention value from the $F_i$, $Z$ is the multi-scale attention weight vector. A soft attention is used across channels to adaptively select different spatial scales, which is guided by the compact feature descriptor $Z_i$. A soft assignment weight is given by

$$att_i = \text{Softmax}(Z_i) = \frac{exp(Z_i)}{\sum_{i=0}^{S-1} exp(Z_i)} \tag{8}$$

where the Softmax is used to obtain the re-calibrated weight $att_i$ of the multi-scale channel, which contains all the location information on the space and the attention weight in channel. By doing this, the interaction between local and global channel attention is realized. Next, the channel attention of feature re-calibration is fused and spliced in a concatenation way, and thus the whole channel attention vector can be obtained as

$$att = att_0 \oplus att_1 \oplus \cdots \oplus att_{S-1} \tag{9}$$

where $att$ represents the multi-scale channel weight after attention interaction. Then, we multiply the re-calibrated weight of multi-scale channel attention $att_i$ with the feature map of the corresponding scale $F_i$ as

$$Y_i = F_i \odot att_i \quad i = 1, 2, 3, \cdots S-1 \tag{10}$$

where $\odot$ represents the channel-wise multiplication, $Y_i$ refers to the feature map that with the obtained multi-scale channel-wise attention weight. The concatenation operator is more effective than the summation due to it can integrally maintain the feature representation without destroying the information of the orginal feature map. In sum, the process to obtain the refined output can be written as

$$Out = \text{Cat}([Y_0, Y_1, \cdots, Y_{S-1}]) \tag{11}$$

As illustrated by the above analysis, our proposed PSA module can integrate the multi-scale spatial information and the cross-channel attention into the block for each feature group. Thus, a better information interaction between local and global channel attention can be obtained by our proposed PSA module.

### 3.3 Network Design

As shown by Figure 4, a novel block named Efficient Pyramid Squeeze Attention (EPSA) block is further obtained by replacing the 3x3 convolution with the PSA module at corresponding positions in the bottelneck blocks of ResNet. The multi-scale spatial information and the cross-channel attention are integrated by our PSA module into the EPSA block. Thus, the EPSA block can extract multi-scale spatial information at a more granular level and develop a long-range channel

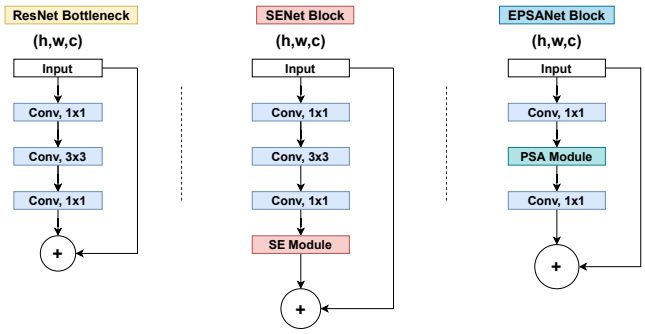


**Fig. 4.** Illustration and comparison of ResNet, SENet, and our proposed EPSANet blocks.

**Table 1.** Network design of the proposed EPSANet.

| Output | ResNet-50 | EPSANet(Small)-50 | EPSANet(Large)-50 |
|---|---|---|---|
| 112×112 | 7×7, 64, stride 2 | | |
| 56×56 | 3×3 max pool, stride 2 | | |
| 56×56 | $\begin{bmatrix} 1\times1, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, & 64 \\ PSA, & 64 \\ 1\times1, & 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, & 128 \\ PSA, & 128 \\ 1\times1, & 256 \end{bmatrix}\times3$ |
| 28×28 | $\begin{bmatrix} 1\times1, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, & 128 \\ PSA, & 128 \\ 1\times1, & 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, & 256 \\ PSA, & 256 \\ 1\times1, & 512 \end{bmatrix}\times4$ |
| 14×14 | $\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, & 256 \\ PSA, & 256 \\ 1\times1, & 1024 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, & 512 \\ PSA, & 512 \\ 1\times1, & 1024 \end{bmatrix}\times6$ |
| 7×7 | $\begin{bmatrix} 1\times1, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, & 512 \\ PSA, & 512 \\ 1\times1, & 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, & 1024 \\ PSA, & 1024 \\ 1\times1, & 2048 \end{bmatrix}\times3$ |
| 1 × 1 | 7×7 global average pool,1000-d fc | | |

dependency. Correspondingly, a novel efficient backbone network named EPSANet is developed by stacking the proposed EPSA blocks as the ResNet style. The proposed EPSANet inherits the advantages of the EPSA block, and thus it has strong multi-scale representation capabilities and can adaptively re-calibrate the cross-dimension channel-wise weight. As shown by Table 1, two variations of the EPSANet, the EPSANet(Small) and EPSANet(Large) are proposed. For the proposed EPSANet(Small), the kernel and group size are respectively set as (3,5,7,9) and (1,4,8,16) in the SPC module. The proposed EPSANet(Large) has a higher group size and is set as (32,32,32,32) in the SPC module.

## 4 Experiments

### 4.1 Implementation Details

For image classification tasks, we employ the widely used ResNet [26] as the backbone model and perform experiments on the ImageNet [27] dataset. The training

configuration is set as the reference in [12, 14, 26]. Accordingly, the standard data augmentation scheme is implemented and the size of the input tensor is cropped to 224×224 by randomly horizontal fliping and normalization. The optimisation is performed by using the stochastic gradient descent (SGD) with weight decay of 1e-4, momentum as 0.9 and a minibatch size of 256. The Label-smoothing regularization [**?**] is used with the coefficient value as 0.1 during training. The learning rate is initially set as 0.1 and is decreased by a factor of 10 after every 30 epochs for 100 epochs in total. For object detection tasks, the ResNet-50 along with FPN [29] is used as the backbone model, we use three representative detectors, Faster RCNN [23], Mask RCNN [4] and RetinaNet [30] on the MS-COCO [31] dataset. The default configuration setting is that the shorter side of the input image is resized to 800. The SGD is used with a weight decay of 1e-4, the momentum is 0.9, and the batch size is 2 per GPU within 12 epochs. The learning rate is set as 0.01 and is decreased by the factor of 10 at the 8th and 11th epochs, respectively. For instance segmentation tasks, we employ the main-stream detection system, Mask R-CNN [4] and also in companion with FPN. The settings of training configuration and dataset are similar to that of the object detection. Finally, all detectors are implemented by the MMDetection toolkit [32], and all models are trained on 8 Titan RTX GPUs.

**Table 2.** Comparison of various attention methods on ImageNet in terms of network parameters(in millions), floating point operations per second (FLOPs), Top-1 and Top-5 Validation Accuracy(%). For a fair comparison, the SKNet* is reduplicated to follow the same training configuration as the proposed EPSANet.

| Network | Backbones | Parameters | FLOPs | Top-1 Acc (%) | Top-5 Acc (%) |
|---|---|---|---|---|---|
| ResNet [26] | ResNet-50 | 25.56M | 4.12G | 75.20 | 92.91 |
| SENet [12] | | 28.07M | 4.13G | 76.71 | 93.70 |
| CBAM [11] | | 28.07M | 4.14G | 77.34 | 93.66 |
| $A^2$-Net[19] | | 33.00M | 6.50G | 77.00 | 93.50 |
| SKNet*[20] | | 26.15M | 4.19G | 77.55 | 93.82 |
| Res2Net+SE [14] | | 28.21M | 4.29G | 78.44 | 94.06 |
| GCNet [1] | | 28.11M | 4.13G | 77.70 | 93.66 |
| Triplet Attention [33] | | 25.56M | 4.17G | 77.48 | 93.68 |
| FcaNet [8] | | 28.07M | 4.13G | 78.52 | 94.14 |
| AANet [34] | | 25.80M | 4.15G | 77.70 | 93.80 |
| ECANet [9] | | 25.56M | 4.13G | 77.48 | 93.68 |
| EPSANet(Small) | | **22.56M** | **3.62G** | 77.49 | 93.54 |
| EPSANet(Large) | | 27.90M | 4.72G | **78.64** | **94.18** |
| ResNet [26] | ResNet-101 | 44.55M | 7.85G | 76.83 | 93.91 |
| SENet [12] | | 49.33M | 7.86G | 77.62 | 94.10 |
| CBAM [11] | | 49.33M | 7.88G | 78.49 | 94.06 |
| AANet [34] | | 45.40M | 8.05G | 78.70 | 94.40 |
| SKNet* [20] | | 45.68M | 7.96G | 78.84 | 94.29 |
| Triplet Attention[33] | | 44.56M | 7.95G | 78.03 | 93.85 |
| ECA-Net [9] | | 44.55M | 7.86G | 78.65 | 94.34 |
| EPSANet(Small) | | **38.90M** | **6.82G** | 78.43 | 94.11 |
| EPSANet(Large) | | 49.59M | 8.97G | **79.38** | **94.58** |

### 4.2 Image Classification on ImageNet

Table 2 shows the comparison results of our EPSANet with prior arts on ResNet with 50 and 101 layers. For the Top-1 accuracy, the proposed EPSANet(Small)-50 achieves a margin of 2.29% higher over the ResNet-50, and using 11.7% fewer parameters and requires 12.1% lower computational cost. Moreover, with almost the same Top-1 accuracy, the EPSANet(Small)-50 can save 54.2% parameter storage and 53.9% computation resources as compared to SENet-101. The EPSANet(Small)-101 outperforms the original ResNet-101 and SENet101 by 1.6% and 0.81% in Top-1 accuracy, and saves about 12.7% parameter and 21.1% computational resources. With the similar Top-1 accuracy on ResNet-101, the computational cost is reduced about 12.7% by our EPSANet(Small)-101 as compared to SRM, ECANet and AANet. What's more, our EPSANet(Large)-50 shows the best performance in accuracy, achieving a considerable improvement compared with all the other attention models. Specifically, the EPSANet(large)-50 outperforms the SENet, ECANet and FcaNet by about 1.93%,1.16% and 0.12% in terms of Top-1 accuracy respectively. With the same number of parameters, our EPSANet(Large)-101 achieves significant improvements by about 1.76% and 0.89% compared to the SENet-101 and CBAM, respectively. In sum, the above results demonstrate that our PSA module has gain a very competitive performance with a much lower computational cost.

### 4.3 Object Detection on MS COCO

As illustrated by Table 3, our proposed models can achieve the best performance for the object detection task. Similar to the classification task on ImageNet, the proposed EPSANet(Small)-50 outperforms the SENet-50 by a large margin with less parameters and lower computational cost. The EPSANet(Large)-50 can achieve the best performance compared with the other attention methods. From the perspective of complexity (in term of parameters and FLOPs), the EPSANet(Small)-50 offers a high competitive performance compared to the SENet50, i.e., by 1.5%, 1.3%, and 1.1%, higher in bounding box $AP$ on the Faster-RCNN, Mask-RCNN, RetinaNet, respectively. What's more, as compared to the SENet50, the EPSANet(Small)-50 can further reducing the number of parameters to 87.5%, 88.3% and 86.4% on Faster RCNN, Mask RCNN and RetinaNet, respectively. The EPSANet(Large)-50 is able to boost the mean average precision by around 4% on the above three detectors as compared with the ResNet-50. It is worth noting that the most compelling performance improvement appears in the measurement of $AP_L$. With almost the same computational complexity, the $AP$ performance can be improved by 1.9% and 1.1% by our proposed EPSANet(Large)-50 on both Faster-RCNN and Mask-RCNN detector, as compared to the FcaNet. The results demonstrate that the proposed EPSANet has good generalization ability and can be easily applied to other downstream tasks.

**Table 3.** Object detection results of different attention methods on COCO val2017.

| Methods | Detectors | Parameters | FLOPs | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|---|
| ResNet-50[26] | Faster R-CNN | 41.53M | 207.07G | 36.4 | 58.4 | 39.1 | 21.5 | 40.0 | 46.6 |
| SENet-50 [12] | | 44.02M | 207.18G | 37.7 | 60.1 | 40.9 | 22.9 | 41.9 | 48.2 |
| ECANet-50 [9] | | 41.53M | 207.18G | 38.0 | 60.6 | 40.9 | 23.4 | 42.1 | 48.0 |
| FcaNet-50 [8] | | 44.02M | 215.63G | 39.0 | 61.1 | 42.3 | 23.7 | 42.8 | 49.6 |
| EPSANet(Small)-50 | | **38.56M** | **197.07G** | 39.2 | 60.3 | 42.3 | 22.8 | 42.4 | 51.1 |
| EPSANet(Large)-50 | | 43.85M | 219.64G | **40.9** | **62.1** | **44.6** | 23.6 | **44.5** | **54.0** |
| ResNet-50[26] | Mask R-CNN | 44.18M | 275.58G | 37.3 | 59.0 | 40.2 | 21.9 | 40.9 | 48.1 |
| SENet [12] | | 46.66M | 261.93G | 38.7 | 60.9 | 42.1 | 23.4 | 42.7 | 50.0 |
| GCNet-50 [1] | | 46.90M | 279.60G | 39.4 | 61.6 | 42.4 | - | - | - |
| ECANet-50 [9] | | 44.18M | 275.69G | 39.0 | 61.3 | 42.1 | 24.2 | 42.8 | 49.9 |
| FcaNet-50 [8] | | 46.66M | 261.93G | 40.3 | 62.0 | 44.1 | 25.2 | 43.9 | 52.0 |
| EPSANet(Small)-50 | | **41.20M** | **248.53G** | 40.0 | 60.9 | 43.3 | 22.3 | 43.2 | 52.8 |
| EPSANet(Large)-50 | | 46.50M | 271.10G | **41.4** | **62.3** | **45.3** | 23.6 | **45.1** | **54.6** |
| ResNet-50[26] | RetinaNet | 37.74M | 239.32G | 35.6 | 55.5 | 38.3 | 20.0 | 39.6 | 46.8 |
| SENet-50 [12] | | 40.25M | 239.43G | 37.1 | 57.2 | 39.9 | 21.2 | 40.7 | 49.3 |
| EPSANet(Small)-50 | | **34.78M** | **229.32G** | 38.2 | 58.1 | 40.6 | **21.5** | 41.5 | 51.2 |
| EPSANet(Large)-50 | | 40.07M | 251.89G | **39.6** | **59.4** | **42.3** | 21.2 | **43.4** | **52.9** |

**Table 4.** Instance segmentation results of different attention networks by using the Mask R-CNN on COCO val2017.

| Network | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|
| ResNet-50 [26] | 34.1 | 55.5 | 36.2 | 16.1 | 36.7 | 50.0 |
| SENet-50 [12] | 35.4 | 57.4 | 37.8 | 17.1 | 38.6 | 51.8 |
| ResNet-50 + 1 NL-block [17] | 34.7 | 56.7 | 36.6 | - | - | - |
| GCNet [1] | 35.7 | 58.4 | 37.6 | - | - | - |
| FcaNet [8] | 36.2 | 58.6 | 38.1 | - | - | - |
| ECANet [9] | 35.6 | 58.1 | 37.7 | 17.6 | 39.0 | 51.8 |
| EPSANet(Small)-50 | 35.9 | 57.7 | 38.1 | 18.5 | 38.8 | 49.2 |
| EPSANet(Large)-50 | **37.1** | **59.0** | **39.5** | **19.6** | **40.4** | 50.4 |

### 4.4 Instance Segmentation on MS COCO

For instance segmentation, our experiments are implemented by using the Mask R-CNN on MS COCO dataset. As illustrated by Table 4, our proposed PSA module outperforms the other channel attention methods by a considerably larger margin. Specifically, our EPSANet(Large)-50 surpass the FcaNet which can offer the best performance in existing methods, by about 0.9% , 0.4% and 1.4% on $AP$, $AP_{50}$ and $AP_{75}$ respectively. These results verified the effectiveness of our proposed PSA module.

## 5 Ablation Studies

In order to provide a comprehensive understanding about the efficiency of our proposed EPSANet. Here, we mainly conduct some ablation experiments to evaluate the performance of each part of the proposed block independently. Such as the effect of kernel size and group size, the benefit of the SPC and SE module, the lightweight performance, and the ability of multi-scale feature representation.

**Table 5.** Accuracy performance with the change of group size on the ImageNet [27] dataset.

| Kernel size | Group size | Top-1 Acc(%) | Top-5 Acc(%) |
|---|---|---|---|
| (3,5,7,9) | (4,8,16,16) | 77.25 | 93.40 |
| (3,5,7,9) | (16,16,16,16) | 77.24 | 93.47 |
| (3,5,7,9) | (1,4,8,16) | **77.49** | **93.54** |

**Table 6.** Accuracy performance with the change of kernel size on the CIFAR-100[35].

| Kernel Size | Group Size | Top-1 Acc(%) |
|---|---|---|
| (3,3,5,5) | (1,4,8,16) | 79.27 |
| (3,5,5,5) | (1,4,8,16) | 79.06 |
| (3,5,5,7) | (1,4,8,16) | 79.67 |
| (3,5,7,9) | (1,4,8,16) | **79.83** |

### 5.1 Effect of the Kernel Size and Group size

Firstly, we explore in detail the combinatorial relationship between the convolution kernel and the group size. The EPSANet(Small)-50 as our baseline model. As shown by Table 5, when the kernel size is fixed as (3,5,7,9), we adjust the group size of different sub-kernel properly. The results show that the best performance can be achieved when the group size is changed as (1,4,8,16). Correspondingly, when the group size is fixed as (1,4,8,16), we adjust the kernel size in different sub-group to explore the best combination relationship. As shown by Table 6, the best performance can be obtained by setting the kernel size as (3,5,7,9). All the above results also verified equation (3).

### 5.2 Effect of the SPC and SE module

Secondly, we conduct an experiment to evulate the benefits coming from the SPC module and the SE module separately. As illustrated by Table 7, the 'SPC' is denote that remove the SE module and only replace the SPC module with the 3x3 convolution in the BottleNeck of the ResNet. The 'SE' is denote that the squeeze size of the SPC module is set as 1, which can be seem as remove the benefits come from the SPC module. The 'SPC+SE' is mean that equipped with the SPC and the SE module. As shown by Table 7, the SPC module and the SE module can bring a more about 0.90% and 0.95% improvement as compared to SENet-50 respectively. The results show that the benefits coming from the SPC module and the SE module are equally important. What's more, equipped with the SPC module and the SE module can achieves a large margin of 1.92% higher accuracy performance over SENet, while using 17.2% fewer parameters.

### 5.3 Effect of the Lightweight performance

Third, as shown by Table 8, the proposed EPSANet can improve the Top-1 accuracy by about 1.76% and 0.98% over the MobileNetV2 and SENet respectively.

**Table 7.** The benefits coming from the SPC module and the SE module on the CIFAR-100 [35] dataset.

| Model | Module | Parameters | Top-1 Acc(%) |
|---|---|---|---|
| SENet-50 [12] | SE | 25.00M | 77.91 |
| EPSANet(Small)-50 | SPC | 20.70M | 78.81 |
| | SE | 23.87M | 78.86 |
| | SPC+SE | 20.71M | **79.83** |

**Table 8.** Comparison of different lightweight attention methods on the ImageNet in terms of network parameters and Top-1 accuracy(%).

| Network | Backbones | Parameters | Top-1 Acc (%) |
|---|---|---|---|
| MobileNetV2[36] | MobileNetV2 | 3.50M | 71.64 |
| SENet[12] | | 3.89M | 72.42 |
| ECA-Net[9] | | 3.50M | 72.56 |
| EPSANet(ours) | | 3.75M | **73.40** |

Meanwhile, as compared to the most competitive model ECANet, the proposed EPSANet also achieves about 0.84% improvement in Top-1 accuracy. Thus, the efficiency and effectiveness of the proposed PSA module for lightweight CNN architectures has verified.

### 5.4 Effect of the Multi-scale Feature Representation

Finally, we mainly compare the proposed EPSANet with several classical multi-scale neural networks. These CNN models have more granular level, deeper and wider architectures, and their results all are copied from the original papers. As shown by Table 9, the proposed EPSANet(Large)-50 outperforms DenseNet-264 [37] and Inception-v3 in terms of Top-1 accuracy, respectively, by about 0.79%, 1.19%. The EPSANet(Large)-50 is very competitive to ResNeXt-101 [38], while the latter one employs more convolution filters and expensive group convolutions. In addition, The proposed EPSANet(Large)-50 is comparable to Res2Net-50 [14], PyConvResNet-50 [13]. All above results demonstrate that the proposed EPSANet has great potential to further improve the ability of multi-scale feature representation of the existing CNN models.

### 5.5 The Visualization Results

For an intuitive demonstration of the intrinsic multi-scale ability, as illustrated by Figure 5, we visualize the class activation mapping (CAM) of the EPSANet(Small)-50 by using Grad-CAM. The visualization results demonstrated that the EPSANet is able to capture richer and more discriminative contextual information for a particular target class.

**Table 9.** Accuracy performance with several classical multi-scale neural networks on the ImageNet dataset.

| Network | Top-1 Acc (%) | Top-5 Acc (%) |
|---|---|---|
| DenseNet-264(k=32) [37] | 77.85 | 93.78 |
| InceptionV3 [28] | 77.45 | 93.56 |
| ResNeXt-101 [38] | 78.80 | 94.40 |
| Res2Net-50 [14] | 77.99 | 93.85 |
| Res2Net-50+SE[14] | 78.44 | 94.06 |
| PyConvResNet-50[13] | 77.88 | 93.80 |
| EPSANet(Large)-50 | 78.64 | 94.18 |

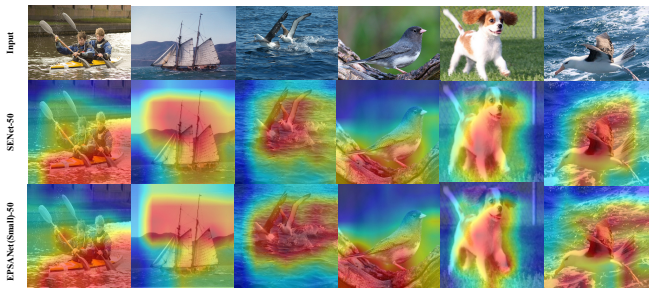


**Fig. 5.** Visualization of GradCAM results. The results are obtained for six random samples from the ImageNet validation set and are compared for SENet50 and EPSANet(Small)-50.

## 6 Conclusion

In this paper, an effective and lightweight attention module named Pyramid Squeeze Attention(PSA) is proposed, which can fully extract the multi-scale spatial information and the important features across dimensions in the channel attention vectors. Correspondingly, the proposed Efficient Pyramid Squeeze Attention(EPSA) block inherits the advantage of the PSA module, which improves the multi-scale representation ability at a more granular level. Extensive qualitative and quantitative experiments demonstrated that the proposed EPSANet surpassed most conventional channel attention methods across a series of computer vision tasks.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China under grants U21A20455, 61972265, 11871348 and 61721002, by the Natural Science Foundation of Guangdong Province of China under grant 2020B1515310008, by the Macao Science and Technology Development Fund under Grant 061/2020/A2, by the Educational Commission of Guangdong Province of China under grant 2019KZDZX1007, the Pazhou Lab, Guangzhou, China.

## References


1. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: International Conference on Computer Vision Workshop (ICCVW), pp. 1971-1980 (2019)
2. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3156–3164 (2017)
3. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: IEEE International Conference on Computer Vision (ICCV), pp. 9626-9635 (2019)
4. He, K., Gkioxari, G., Doll´ar, P., Girshick, R.: Mask r-cnn. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 386-397 (2020)
5. Zhong, Z., Lin, Z.Q., Bidart, R., Hu, X., Daya, I.B., Li, Z., Zheng, W.S., Li, J., Wong, A.: Squeeze-and-attention networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13062–13071 (2020)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (ECCV), pp. 833–851 (2018)
7. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6230–6239 (2017)
8. Qin, Z., Zhang, P., Wu, F., Li, X.: Fcanet: Frequency channel attention networks. In: International Conference on Computer Vision (ICCV), pp. 763-772 (2021)
9. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11531–11539 (2020)
10. Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: Bottleneck attention module. In: British Machine Vision Conference(BMVC) (2018)
11. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132–7141 (2018)
13. Duta, I.C., Liu, L., Zhu, F., Shao, L.: Pyramidal convolution:rethinking convolutional neural networks for visual recognition (2020)
14. Gao, S.H., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.: Res2net: A new multi-scale backbone architecture. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 2, pp. 652-662 (2021)
15. Yuan, P., Lin, S., Cui, C., Du, Y., Guo, R., He, D., Ding, E., Han, S.: Hs-resnet: Hierarchical-split block on convolutional neural network (2020)
16. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3141–3149 (2019)
17. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7794–7803 (2018)
18. Gao, Z., Xie, J., Wang, Q., Li, P.: Global second-order pooling convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3019–3028 (2019)

19. Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: A2-nets: Double attention networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 350–359 (2018)
20. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–519 (2019)
21. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Muller, J., Manmatha, R., Li, M., Smola, A.: Resnest: Split-attention networks (2020)
22. Najibi, M., Samangouei, P., Chellappa, R., Davis, L.S.: Ssh: Single stage headless face detector. In: IEEE International Conference on Computer Vision (ICCV), pp. 4885–4894 (2017)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149 (2017)
24. Zhao, J.X., Cao, Y., Fan, D.P., Cheng, M.M., Li, X.Y., Zhang, L.: Contrast prior and fluid pyramid integration for rgbd salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3922–3931 (2019)
25. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: International Conference on Machine Learning (ICML), pp. 807–814 (2010)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
27. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A largescale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255 (2009)
28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016)
29. Lin, T.Y., Doll´ar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
30. Lin, T.Y., Goyal, P., Girshick, R., He, K., Doll´ar, P.: Focal loss for dense object detection. In: IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017)
31. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV), pp. 740–755 (2014)
32. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T.h., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: Mmdetection: Open mmlab detection toolbox and benchmark (2019)
33. Misra, D., Nalamada, T., Arasanipalai, A.U., Hou, Q.: Rotate to attend: Convolutional triplet attention module. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 3138–3147 (2021)
34. Bello, I., Zoph, B., Le, Q., Vaswani, A., Shlens, J.: Attention augmented convolutional networks. In: IEEE International Conference on Computer Vision (ICCV), pp. 3285–3294 (2019)
35. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases (2009)

36. Mark, S., Andrew, G.H., Zhu, M., Andrey, Z., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520 (2018)
37. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261-2269 (2017)
38. Xie, S., Girshick, R., Doll´ar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987–5995 (2017)