Nonlocal spatial attention module for image classification

Bingling Chen¹, Yan Huang², Qiaoqiao Xia¹ and Qinglin Zhang¹

Abstract
To enhance the capability of neural networks, research on attention mechanism have been deepened. In this area, attention modules make forward inference along channel dimension and spatial dimension sequentially, parallelly, or simultaneously. However, we have found that spatial attention modules mainly apply convolution layers to generate attention maps, which aggregate feature responses only based on local receptive fields. In this article, we take advantage of this finding to create a nonlocal spatial attention module (NL-SAM), which collects context information from all pixels to adaptively recalibrate spatial responses in a convolutional feature map. NL-SAM overcomes the limitations of repeating local operations and exports a 2D spatial attention map to emphasize or suppress responses in different locations. Experiments on three benchmark datasets show at least 0.58% improvements on variant ResNets. Furthermore, this module is simple and can be easily integrated with existing channel attention modules, such as squeeze-and-excitation and gather-excite, to exceed these significant models at a minimal additional computational cost (0.196%).

Keywords
Convolutional neural network, nonlocal, attention module, image classification, computer vision, object recognition and classification, vision systems

Date received: 23 March 2020; accepted: 8 June 2020

Topic Area: Vision Systems
Topic Editor: Antonio Fernandez-Caballero
Associate Editor: Wendy Flores-Fuentes

Introduction
By interleaving a series of convolutional layers with non-linear activation functions and downsample operators, convolutional neural networks (CNNs)¹ are able to produce robust representations that capture hierarchical patterns and attain global theoretical receptive field. Thus, CNNs become the paradigm of choice in many computer vision applications, such as image classification,²⁻⁵ object detection,⁶ semantic segmentation,⁷ and regression.⁸⁻⁹ In recent years, attention mechanisms have been a new remedy for feature recalibration by capturing contextual long-range interactions. The attention mechanism started from the introduction of an attention module to draw global dependencies of inputs in neural machine translation,¹⁰ and then, the landmark works about self-attention modules¹¹ sets a new standard in this field.

Self-attention mechanism is to measure the compatibility of the query and key content pairwise relations. In this field, one approach is that nonlocal network (NLNet)¹² presents a self-attention map to model the correspondence from all positions to each query position. Meanwhile, simplified attention models use a query-independent attention map for all query positions. In the recently proposed convolutional block attention module (CBAM),¹³ a single

¹ College of Physical Science and Technology, Central China Normal University, Wuhan, China
² Algorithm group, ZMVision Technology, Wuhan, China

Corresponding author:
Qinglin Zhang, College of Physical Science and Technology, Central China Normal University, Wuhan 430072, China.
Email: zhangqinglin@mail.ccnu.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
spatial attention map is multiplied back to the channel-attention tuned feature maps for adaptive feature refinement. Our nonlocal spatial attention module (NL-SAM) builds on the benefits of NLNet with effective modeling on global contextual information and CBAM with an efficient attention map generation. NL-SAM incorporates three interdependent operations: context collecting, transformation, and distribution. Context collecting performs feature aggregation to obtain highly compressed global information in spatial dimension. Transformation provides a nonlinear way to recalibrate feature responses. Then, attention map is distributed to each location in the convoluted feature maps.

Our NL-SAM is general and efficient in terms of added parameters, it can be integrated into any CNN architectures individually while maintaining end-to-end trainable or existing channel attention modules with negligible overhead to serve as the complementary attention. We incorporate NL-SAM within ResNet\(^4\) to validate through experiments on CIFAR-10, CIFAR-100, and ImageNet-1K classification datasets. In particular, as shown in Figure 1, deep CNNs embedded with our NL-SAM introduce very few additional computations while bringing notable performance gain. For example, for ResNet50 with 25.53M parameters and 6.995G floating point operations (FLOPs), the additional parameters and computations of NL-SAM are only 0.03M and 0.007G FLOPs, but the accuracy has been improved by 0.58%. And gather-excite network (GENet)\(^14\) combined with NL-SAM reaches the highest accuracy (75.6\%) with only 0.085% additional FLOPs (more details are given in Table 7).

The rest of this article is organized as follows. The second section analyzes related works about attention mechanisms. The third section describes the architecture of the proposed NL-SAM and analyzes the relationship between NL-SAM and other attention modules. The fourth section verifies the effectiveness of NL-SAM throughout extensive experiments with various baseline models on multiple benchmarks. The fifth section concludes the article.

Related works

Convolution network architecture

The convolution layer has been the dominant visual feature extractor in computer vision. Recent advances of convolution networks focus on capturing long-range dependencies.\(^3,4\) ResNet\(^4\) solves the degradation problem caused by the increased depth of neural networks, thus, it is able to deliver information between distant positions by simply increasing the network depth. After that, another important direction is to modify the spatial scope for aggregation by enlarging the receptive field, such as atrous/dilated convolution.\(^7\)

Figure 1. Comparison of various attention modules embedded in ResNet50 in terms of accuracy and FLOPs on ImageNet, and the size of circles indicates parameter numbers. FLOP: floating point operation.

Attention models

In vision, the key and query refer to visual elements, such as image pixels. Regular convolution can be deemed as a special instantiation of spatial attention mechanism. Given a query element, predetermined positional offsets of key elements are sampled. In the recently proposed attention mechanisms, there are two major patterns as follows.

Query-independent attention models. These models are irrelevant to the query content, and they only capture salient key contents, which should be focused on for the task. By computing a global attention map and sharing this map for all query positions, these models are very efficient. For example, squeeze-and-excitation network (SENet)\(^15\) and GENet\(^14\) perform rescaling to different channels to recalibrate the channel dependency. However, they miss the spatial axis, which is also important for inferring accurate attention maps. Bottleneck attention module (BAM)\(^16\) and CBAM\(^13\) introduce spatial attention using convolution in a similar way with a channel attention mechanism. However, in these spatial attention models, feature transformation is performed by convolution, which results in suboptimal effect for global context exploitation in CNNs.

Query-specific attention models. Motivated by their success in natural language processing (NLP) tasks, self-attention mechanisms have also been employed in computer vision applications, such as image recognition,\(^17–21\) relational reasoning among objects,\(^22,23\) image segmentation,\(^17,24\) scene parsing,\(^22\) and video recognition.\(^26\) Wang et al.\(^27\) have proposed residual attention network, which uses an hourglass module to generate 3D attention maps for intermediate features. As attention maps are computed for each query position, the time and space complexity are both quadratic to the number of positions. To decrease the large computational
overheads caused by the heavy map generation process, dual attention network\textsuperscript{20} appends two parallel 2D and 1D attention modules on top of dilated fully convolutional networks (FCN).\textsuperscript{28} However, they all have an obvious shortcoming—self-attention models contain a large number of matrix multiplication operations, which increases the computational burden.

**Proposed method**

In this section, we first overview two major attention modules NLNet\textsuperscript{12} and CBAM,\textsuperscript{13} and conclude a concise formula that summarizes these attention models. Then, we introduce the NL-SAM and describe its key components. In addition, we analyze the relationship between our NL-SAM and other attention modules.

**Overview of attention modules**

A representative query-independent attention model is CBAM,\textsuperscript{13} which is formulated as

\[
Z_0 = \frac{M_c(X)}{C} \times X \quad \text{(3)}
\]

\[
Z = \frac{M_s(Z)}{C} \times Z' \quad \text{(2)}
\]

where \(X \in \mathbb{R}^{W \times H \times C}\) is the input feature block (\(W\), \(H\), and \(C\) are width, height, and channel number, respectively), and \(\otimes\) denotes element-wise multiplication, \(Z\) is the final refined output. CBAM first computes the channel attention

\[
M_c(X) = \sigma \left( W_1 \left( W_0 \sum_{\forall j} \frac{1}{N_j} X_j \right) + W_1 \left( W_0 \max_{\forall j} (X_j) \right) \right) \quad \text{(4)}
\]

In this function, \(W_1\) and \(W_0\) are multilayer perceptron weights, \(\forall j\) enumerates all positions in the \(j\)th feature map, and \(N_j\) is the number of positions in this feature map, \(\max(\cdot)\) calculates the largest value of the inputs, and \(\sigma\) refers to the sigmoid function. Based on the channel attention output \(Z'\), the spatial attention model is calculated as

\[
M_s(Z') = \sigma \left( W_7 \times 7 \left( \left( \frac{1}{C} \sum_{\forall j} Z'_j, \max_{\forall j} (Z'_j) \right) \right) \right) \quad \text{(5)}
\]

where \(\forall j\) denotes \(1 \leq j \leq C\) and \(W_7\times7\) denotes convolution with kernel size 7.

NLNet,\textsuperscript{12} a typical query-specific attention module, can be expressed as

\[
Z_i = X_i + W \sum_{\forall j} \frac{f(X_i, X_j)}{C(X)} (W_1 \times 1 X_j) \quad \text{(6)}
\]

where \(i\) is the index of query positions and \(\forall j\) enumerates all possible positions. \(f(X_i, X_j)\) denotes the relationship between positions \(i\) and \(j\), and \(C(X)\) is a normalization factor.

After simplifying these attention models, a general attention modeling framework can be summarized as the following formula

\[
Z = F(X, \delta_{\forall j}(X_j))
\]

i. \(\delta_{\forall j}(X_j)\) represents a context collecting module that aggregates features at all locations to obtain global context features through weighted calculations and \(\forall j\) represents the traversal value of the channel dimension or spatial dimension;

ii. \(\delta(\cdot)\) denotes the nonlinear feature conversion method;

iii. \(F(\cdot, \cdot)\) denotes the attention distribution function that broadcasts the global context features to each position.

**Nonlocal spatial attention module**

The structure of NL-SAM is shown in Figure 2, which can be roughly divided into three parts: context collecting,
| Ratio | Top-1 accuracy (%) | Top-5 accuracy (%) | Params (M) | FLOPs (G) |
|-------|-------------------|-------------------|------------|-----------|
| 4     | 74.37             | 92.32             | 11.830     | 0.552     |
| 8     | 74.62             | 92.36             | 6.795      | 0.53      |
| 16    | 74.41             | 92.68             | 4.278      | 0.521     |
| ResNet110 | 73.86         | 92.02             | 1.737      | 0.511     |

FLOP: floating point operation.
The bold values indicate the highest accuracy in the comparison experiment.

Table 1. Accuracy (%) on CIFAR-100 test set by ResNet110 with different reduction ratio r.

Table 2. Effect of inserting NL-SAM at different stages with enumerated values of k on ResNet110 with r = 8.

| Network       | Kernel k | Top-1 accuracy (%) | Top-5 accuracy (%) | Params (M) | FLOPs (G) |
|---------------|----------|-------------------|-------------------|------------|-----------|
| ResNet110     | 0        | 73.86             | 92.02             | 1.737      | 0.511     |
| ResNet110 + NL-SAM stage1 | 4    | 74.19             | 92.20             | 1.756      | 0.511     |
| ResNet110 + NL-SAM stage2 | 2    | 74.64             | 92.32             | 1.756      | 0.511     |
| ResNet110 + NL-SAM stage3 | 2    | 74.73             | 92.60             | 1.738      | 0.511     |
| ResNet110 + NL-SAM all    | 2    | 74.74             | 92.71             | 2.057      | 0.513     |

NL-SAM: nonlocal spatial attention module; FLOP: floating point operation.
The bold values indicate the highest accuracy in the comparison experiment.

In detail, our NL-SAM consists of

i. channel reduction for information aggregation and local average pooling on feature map for context collecting, as \( \theta_k(\cdot) \) defined in equation (9);

ii. bottleneck transformation capture pixelwise dependencies, as \( \delta_c(\cdot) \) defined in equation (10);

iii. matrix multiplication for feature fusion, as \( F(\cdot, \cdot) \) defined in equation (11).

The first part is context collecting for global information aggregation. For simplicity, channel average pooling is used here. For a given feature block \( X \in \mathbb{R}^{W \times H \times C} \), the output \( X_{cp} \) is computed as

\[
X_{cp}(i,j) = \frac{1}{C} \sum_{m=1}^{C} X(i,j,m)
\]

(8)

On the basis of channel compression, we further utilize average pooling to aggregate the channel-pooled feature maps. Experimental results in Table 2 demonstrate that pooling of adjacent pixels in early stages where NL-SAM inserted provides translation invariance needed for classification. The final aggregated feature map \( X_o \) is calculated by

\[
X_o = \frac{1}{k} \sum_{m=i/k}^{(i'+1)/k} X_{cp}(m,n)
\]

(9)

where \( k \) is a hyperparameter of the pooling operation with different values at different stages, \( i' \) and \( j' \) are the traversal values of the context collecting map, where \( 0 \leq i' \leq \frac{W}{k} \) and \( 0 \leq j' \leq \frac{H}{k} \).

The second part—feature transformation—has the largest number of parameters. Thus, a bottleneck transformation module is applied, which significantly reduces the number of parameters from \( WH \cdot WH \) to \( 2 \cdot WH \cdot \frac{WH}{r} \), where \( r \) is the bottleneck ratio and \( \frac{WH}{r} \) denotes the

neuron number of the bottleneck. With default reduction ratio set to \( r = 8 \), the number of parameters and computational overheads for transformation module can be reduced to \( \frac{1}{8} \). More results on different values of bottleneck ratio \( r \) are given in Table 1.

Then, the 2D pooled feature map \( X_o \in \mathbb{R}^{\frac{W}{k} \times \frac{H}{k}} \) is transformed into one dimension \( X'_{o} \in \mathbb{R}^{\frac{WH}{r}} \). Each point in the feature map is treated as an independent node, the trainable weight matrix learns the “importance” or saliency of each point corresponds to the rest in the feature map by backpropagation. The weight matrix is query-sensitive and can learn exclusive relationships since every query has its own attention map. This is important because of the pooling operation in the original feature map and the bottleneck reduction operation in this part. Inspired by SENet, full connection acts as an instantiation here

\[
X_s = \text{sigmoid}(\text{fc}((\text{fc}(X'_o))))
\]

(10)

where “fc” means full connection. Finally, the 1D results are restored to 2D, namely, the transformation results \( X_s \in \mathbb{R}^{\frac{WH}{r}} \).

The main calculation cost of NL-SAM is introduced by this part. The FLOPs are \( \frac{WH}{r} \), which are determined by the width \( W \) and height \( H \) of the input feature map, and pooling kernel size \( k \) and bottleneck reduction ratio \( r \).

In the last part, fusion function distributes the global context to the features of each position. While the transformed feature map has a width/height that is \( 1/k \) of the original size, experimental results in Table 3 draw a conclusion that nearest neighbor interpolation is the best choice to restore the size. The finally refined feature map is obtained by

\[
X'_s = \text{resize}(X_s)
\]

(11)

\[
Z = X \otimes X'_s
\]

(12)

where \( \otimes \) is the element-wise multiplication.
where $\otimes$ denotes elementwise multiplication, and the same sized 2D spatial attention feature map $X_{d2} \in \mathbb{R}^{W \times H}$ is to emphasize or suppress responses in different locations.

**Relationship to other attention modules**

We make a theory comparison for deeper analysis on the relationship to other attention modules. To aggregate global context from all positions, SENet\(^{15}\) utilizes global average pooling, which assumes that every point in a feature map contributes the same to the whole, there is no spatial attention in this module, so it can serve as the channel aggregation block of our NL-SAM. GENet\(^{14}\) and BAM\(^{16}\) utilize depthwise global convolution or dilated convolution to obtain a large receptive field, since convolution finally sums up as a single result, it is only one point on our 2D attention map. In the fusion stage, SENet and GENet utilize rescaling to recalibrate a channel as a whole unit by one single parameter; BAM\(^{16}\) utilizes broadcast elementwise addition; CBAM\(^{13}\) is almost the same as us, the most sophisticated way is elementwise multiplication.

Our NL-SAM follows the NLNet\(^{12}\) by acquiring a query-sensitive attention map, but we need to specify that there are two prominent features in this block. Firstly, we use dimension reduction to decompose channel attention and spatial attention. Secondly, we use the combination of $k \times k$ pooling and upsampling to reduce the calculation in global information collection and to maintain translation invariance. And, moreover, NL-SAM is a simplified version and not referred to self-attention but salience-attention.

**Experiment and analysis**

In this section, we evaluate the performance of the proposed NL-SAM on a series of benchmark datasets, including CIFAR-10, CIFAR-100, and ImageNet.\(^{29}\) Our experiments contain three parts. Firstly, ablation experiments are conducted to determine the optimal value of hyperparameters. Given limited computation resources, CIFAR-10 and CIFAR-100 are selected as experimental datasets. Our module surpasses ResNet on CIFAR datasets both in efficiency and accuracy. Secondly, we analyze the reciprocal relation of NL-SAM with other attention modules. Finally, to be more convincing, we compare classification results with ResNet50 variants on ImageNet. And evaluate original ResNets based on the official code of TensorFlow-Slim image classification model library.\(^{30}\) Some parameters are modified due to limited computing resources, especially the batch size. For fair comparison, all the relevant attention models are reimplemented by TensorFlow-Slim.

**CIFAR and analysis**

**Implementation details.** We follow the simple data augmentation strategy in the literature\(^4\) for training: images are randomly flipped horizontally and zero-padded on each side with four pixels before taking a random $32 \times 32$ crop, mean and standard deviation normalization are also applied. ResNet110 and ResNet164 preactivation residual models\(^5\) are trained on one RTX2080ti, batch size is set to 128, momentum and weight decay are 0.9 and 0.0003 and the same weight initialization as the reference.\(^5\) The initial learning rate is set to 0.1 with warming up at the first 400 iterations, divided by 10 at 120, 190 of 250 epochs in total.

**Reduction ratio.** The bottleneck compression rate $r$ is intended to reduce redundancy in parameters. The value of average pooling kernel size $k$ is fixed to 0, different values are set to determine the optimal value of $r$. As given in Table 1, $r = 8$ achieves a good balance between accuracy and complexity. So, this value is used in the rest of the experiments.

**Effect on different inserting stages and pooling kernel size.** The influence of NL-SAM is investigated on different stages (here the term “stage” is the same as defined in the literature\(^4\)) based on different inserting stages and different values of $k$; the accuracy, computational cost, and parameter numbers, as given in Table 2.

---

**Table 3. Accuracy(%) of interpolation on stage 2 of ResNet110 + NL-SAM with $k = 2$.**

| ResNet110 + NL-SAM (stage 2, $k = 2$) | Top-1 accuracy (%) | Top-5 accuracy (%) | Params (M) | FLOPs (G) |
|-------------------------------------|--------------------|--------------------|------------|----------|
| Nearest neighbor                    | 74.64              | 92.32              | 1.756      | 0.511    |
| Bilinear                            | 74.43              | 92.24              | 1.756      | 0.511    |

NL-SAM: non-local spatial attention module; FLOP: floating point operation.

The bold values indicate the highest accuracy in the comparison experiment.
Based on the results listed in Table 2 and the corresponding Figure 3, we have the following observations: (1) deeper stage which has more semantic information benefits more from NL-SAM; (2) insertion at all stages are not mutually redundant, they can further bolster the accuracy; (3) pooling is not only to reduce the calculation cost but also to keep translation invariant in classification, since our module is position-sensitive, weight sharing within the pooling kernel is necessary to counteract this.

At the same time, the interpolation experiment is conducted in stage 2 with pooling kernel size of 2. The results of Table 3 verify that nearest-neighbor interpolation is the best choice. In fact, due to the previous average pooling of the feature map, the same weights are naturally shared within the \( k \times k \) block in \( X' \) (equation (11)).

### Integration with different architectures.

We evaluate our NL-SAM on the popular ResNet backbones, plain ResNet110\(^5\) and ResNet164\(^5\) with bottleneck. According to the above experiments, the structure and hyperparameters are determined, as given in Table 4. NL-SAM embedded in stage 1 of the network consumes more resources, but the effect is inferior to the later stages. Thus, it is the best choice to insert NL-SAM into stage 2 and stage 3 with \( k \) set to 2, and the results are reported in Table 5. It can be found that NL-SAM consistently improves performance across different depths with a small increase in FLOPs.

### Combination with other attention modules.

Comparative experiments are conducted on CIFAR-100 by integrating NL-SAM with SENet and GENet to serve as complementary spatial attention. Average pooling is used in SENet, global depthwise convolution is used in GENet, and bottleneck reduction ratio is set to 4 with sigmoid activation in both SENet and GENet. In Figure 4, the comparison of parameter numbers and accuracy of different models are explicit. It shows that NL-SAM can further improve the accuracy of SENet and GENet by about 0.2% with less than 0.002M parameters added. Furthermore, we replace spatial attention module in CBAM with NL-SAM. In the third column of the figure, we can see that this replacement variant outperforms the original CBAM by 0.32%. It suggests that the designed query-sensitive spatial attention in NL-SAM is superior to query-independent spatial attention in CBAM.

### ImageNet and analysis

**Implementation details.** This dataset contains 1.2 million training images and 50k validation images, each is cropped to \( 224 \times 224 \) pixels. ResNet50 is chosen as a base model.
and trained from random initialization. SGD is used with momentum as 0.9 to optimize, batch size is set to 128. The initial learning rate is set to 0.1 and is reduced by a factor of 10 for every 30 epochs, all models are trained for 100 epochs.

**Classification results on ImageNet.** Based on the ablation experiments on CIFAR, the hyperparameters of ResNet50 for ImageNet are defined in Table 6. Due to the large feature map size and insertion stage position study on CIFAR, we add NL-SAM in stages 3, 4, and 5.

We further analyze the effectiveness of NL-SAM by the splitting and combination experiments. The results of the final models are given in Table 7. It can be found that our module is able to effectively improve the accuracy even on large datasets. Compared with SENet and GENet, NL-SAM is slightly inferior but still has a notable improvement on ResNet50 baseline. The best result is produced when NL-SAM is integrated with GENet. CBAM seems to be unsteady, which does not reach the baseline result, but when integrated with NL-SAM, the accuracy is improved by 0.22\%. These experiments verify the consistent improvement of NL-SAM when integrated with channel attention modules. NL-SAM exhibits the ability to use without bells and whistles.

**Analysis and discussion**

We compare the visualization results of NL-SAM inserted to ResNet50 or combined with other modules. The grad-CAM visualization is calculated for the last convolution layer outputs, ground-truth label is shown on the top of each input image and \( P \) denotes the softmax score of each network for the predicted class. As shown in Figure 5, according to \( P \)'s score, ResNet50 misclassifies siamese cat and obtains the lowest score. ResNet50 + GE + NL-SAM always gets the highest score; ResNet50 + NL-SAM and ResNet50 + GENet both have their own advantages. We

---

**Table 6.** Architecture of ResNet50 and the hyperparameters \( r \) and \( k \) of NL-SAM.

| Output size | ResNet50 | NL-SAM \((r, k)\) |
|------------|----------|------------------|
| 112 \( \times \) 112 stage 1 | Conv, 7 \( \times \) 7, stride 2 | — |
| 56 \( \times \) 56 stage 2 | Maxpooling, 3 \( \times \) 3, stride 2 | — |
| 28 \( \times \) 28 stage 3 | | |
| \( k = 2 \) 14 \( \times \) 14 stage 4 | Conv, 3 \( \times \) 3, 256, \( r = 8 \), \( k = 27 \) | |
| \( k = 2 \) 7 \( \times \) 7 stage 5 | Conv, 3 \( \times \) 3, 512, \( r = 8 \), \( k = 21 \) | |
| \( k = 0 \) 1 \( \times \) 1 | cGlobal avgpooling 1000-d, softmax | — |

NL-SAM: nonlocal spatial attention module.
can clearly see that NL-SAM integrated networks have a good incentive or suppression in some regions. For example, they focus more precisely on the head of the merganser and pay less attention to the water, and attention on the whole body of siamese cat shows the benefit of global visual field.

**Conclusion**

We propose the NL-SAM, which is a new method to enhance the representation ability of the network. Through modeling on global contextual information, our module learns where to focus or suppress, and refines intermediate features. To verify its effectiveness, extensive experiments were conducted on three benchmark datasets. The results show that NL-SAM improves classification accuracy by 0.98% on CIFAR-10, 1.01% on CIFAR-100 when embedded in ResNet110, and 0.58% on ImageNet when embedded in ResNet50. What is more, the combination of NL-SAM and GENet improves accuracy on ImageNet by 1.05% with only 0.038G FLOPs added. For future works, we plan to devise a flexible way to switch between NL-SAM and convolution in a layer.

**Table 7.** Classification accuracy (%) on ImageNet validation set with embedded NL-SAM, CBAM in ResNet50, SENet50, and GENet50.

| Network                  | Top-1 accuracy (%) | Top-5 accuracy (%) | Params (M) | FLOPs (G) |
|--------------------------|--------------------|--------------------|------------|-----------|
| ResNet50                 | 74.50              | 91.9               | 25.530     | 6.995     |
| ResNet50 + NL-SAM        | 75.08              | 92.11              | 25.580     | 7.002     |
| ResNet50 + SE            | 75.13              | 92.38              | 28.061     | 7.015     |
| ResNet50 + SE + NL-SAM   | 75.25              | 92.38              | 28.106     | 7.020     |
| ResNet50 + GE            | 75.15              | 92.45              | 32.500     | 7.027     |
| ResNet50 + GE + NL-SAM   | 75.55              | 92.49              | 32.545     | 7.033     |
| ResNet50 + CBAM          | 74.08              | 92.04              | 28.063     | 7.030     |
| ResNet50 + CBAM-ch + NL-SAM | 74.72          | 92.04              | 28.106     | 7.026     |

NL-SAM: nonlocal spatial attention module; FLOP: floating point operation. The bold values indicate the highest accuracy in the comparison experiment.

**Figure 5.** Grad-CAM visualization results. Grad-CAM: gradient - class activation map.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially supported by the National Natural Science Foundation of China [Grant Nos. 61701351] and the Fundamental Research Funds for the Central Universities of CCNU [No. CCNU16A05028].

ORCID iDs
Bingling Chen  https://orcid.org/0000-0002-6009-1640
Qinglin Zhang  https://orcid.org/0000-0001-6023-6734

References
1. LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proc IEEE 1998; 86(11): 2278–2324.
2. Krizhevsky A, Sutskever I, and Hinton GE. ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems (NIPS), Lake Tahoe, Nevada, USA, 3–6 December 2012, pp. 1097–1105. Red Hook, NY: Curran Associates Inc.
3. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556 2014.
4. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778. Piscataway, NJ: IEEE.
5. He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks. In: European conference on computer vision, Amsterdam, The Netherlands, October 11–14, 2016, pp. 630–645, Berlin: Springer.
6. Ren S, He K, Girshick R, et al. Faster r-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, Montreal, Quebec, Canada, 7–10 December 2015, pp. 91–99, Cambridge, MA: MIT Press.
7. Chen LC, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 2017; 40(4): 834–848.
8. Trujillohernandez G, Rodriguezquinonez JC, Ramirezhernandez LR, et al. Accuracy improvement by artificial neural networks in technical vision system. In: Annual conference of IEEE industrial electronics society, Lisbon, Portugal, Portugal, 14–17 October 2019, pp. 5572–5577. IEEE.
9. Rodriguezquinonez JC, Sergiyenko O, Floresfuentes W, et al. Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods’ approach. Opto-electron Rev 2017; 25(1): 24–32.
10. Luong MT, Pham H, and Manning CD. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:150804025 2015.
11. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Advances in neural information processing systems, Long Beach, CA, USA, 4–9 December 2017, pp. 5998–6008. Red Hook, NY: Curran Associates Inc.
12. Wang X, Girshick R, Gupta A, et al. Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018, pp. 7794–7803. Piscataway, NJ: IEEE.
13. Woo S, Park J, Lee JY, et al. CBAM: convolutional block attention module. In: Proceedings of the european conference on computer vision (ECCV), Munich, Germany, September 8–14, 2018, pp. 3–19. Cham: Springer.
14. Hu J, Shen L, Albanie S, et al. Gather-excite: exploiting feature context in convolutional neural networks. In: Advances in neural information processing systems, Montreal, Quebec, Canada, 3–8 December 2018, pp. 9401–9411. Red Hook, NY: Curran Associates Inc.
15. Hu J, Shen L, and Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Wellington, New Zealand, June 2018, pp. 7132–7141. Piscataway, NJ: IEEE.
16. Park J, Woo S, Lee JY, et al. BAM: bottleneck attention module. arXiv preprint arXiv:180706514 2018.
17. Huang Z, Wang X, Huang L, et al. CCNet: criss-cross attention for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision, 2019, pp. 603–612, Washington, DC: IEEE.
18. Gu J, Hu H, Wang L, et al. Learning region features for object detection. In: Proceedings of the european conference on computer vision (ECCV), January 2018, pp. 381–395. Cham: Springer.
19. Zhao H, Zhang Y, Liu S, et al. PSANet: point-wise spatial attention network for scene parsing. In: Proceedings of the european conference on computer vision (ECCV), Munich, Germany, September 8–14, 2018, pp. 267–283. Cham: Springer.
20. Fu J, Liu J, Tian H, et al. Dual attention network for scene segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Long Beach, CA, USA, 15–20 June 2019, pp. 3146–3154. Piscataway, NJ: IEEE.
21. Cao Y, Xu J, Lin S, et al. GCNet: non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE international conference on computer vision workshops, Seoul, Korea, 27–28 October 2019. Piscataway, NJ: IEEE.
22. Battaglia P, Pascanu R, Lai M, et al. Interaction networks for learning about objects, relations and physics. In: Advances in neural information processing systems, Barcelona, Spain, 5–10 December 2016, pp. 4502–4510. Red Hook, NY: Curran Associates Inc.
23. Santoro A, Raposo D, Barrett DG, et al. A simple neural network module for relational reasoning. In: Advances in neural information processing systems, Long Beach, CA, USA, 4–9 December 2017, pp. 4967–4976. Red Hook, NY: Curran Associates Inc.

24. Chen LC, Yang Y, Wang J, et al. Attention to scale: scale-aware semantic image segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016, pp. 3640–3649. Piscataway, NJ: IEEE.

25. Yuan Y and Wang J. OCNet: object context network for scene parsing. arXiv preprint arXiv:180900916 2018.

26. Ge H, Yan Z, Yu W, et al. An attention mechanism based convolutional LSTM network for video action recognition. Multimedi Tools Appl 2019; 78(14): 20533–20556.

27. Wang F, Jiang M, Qian C, et al. Residual attention network for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017, pp. 3156–3164. Piscataway, NJ: IEEE.

28. Long J, Shelhamer E, and Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, USA, 7–12 June 2015, pp. 3431–3440. Piscataway, NJ: IEEE.

29. Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition (CVPR), Miami, FL, USA, 20–25 June 2009, pp. 248–255. Piscataway, NJ: IEEE.

30. N Silberman SG. TensorFlow-slim image classification model library. https://github.com/tensorflow/models/tree/master/research/slim, 2016 (accessed April 2018).

31. Selvaraju RR, Cogswell M, Das A, et al. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, Venice, Italy, 22–29 October 2017, pp. 618–626. Washington, DC: IEEE.