applied sciences

MDPI

Article

A New Multi-Scale Convolutional Model Based on Multiple Attention for Image Classification

Yadong Yang [1], Chengji Xu [1], Feng Dong [1,2] and Xiaofeng Wang [1,*]

1 College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China; yangyadong03@stu.shmtu.edu.cn (Y.Y.); xuchengji@stu.shmtu.edu.cn (C.X.); dongfeng@stu.shmtu.edu.cn (F.D.)

2 College of Information Engineering, Shaoyang University, Shaoyang 422000, China

* Correspondence: xfwang@shmtu.edu.cn

Received: 12 October 2019; Accepted: 17 December 2019 ; Published: 20 December 2019

Abstract: Computer vision systems are insensitive to the scale of objects in natural scenes, so it is important to study the multi-scale representation of features. Res2Net implements hierarchical multi-scale convolution in residual blocks, but its random grouping method affects the robustness and intuitive interpretability of the network. We propose a new multi-scale convolution model based on multiple attention. It introduces the attention mechanism into the structure of a Res2-block to better guide feature expression. First, we adopt channel attention to score channels and sort them in descending order of the feature's importance (Channels-Sort). The sorted residual blocks are grouped and intra-block hierarchically convolved to form a single attention and multi-scale block (AMS-block). Then, we implement channel attention on the residual small blocks to constitute a dual attention and multi-scale block (DAMS-block). Introducing spatial attention before sorting the channels to form multi-attention multi-scale blocks(MAMS-block). A MAMS-convolutional neural network (CNN) is a series of multiple MAMS-blocks. It enables significant information to be expressed at more levels, and can also be easily grafted into different convolutional structures. Limited by hardware conditions, we only prove the validity of the proposed ideas through convolutional networks of the same magnitude. The experimental results show that the convolution model with an attention mechanism and multi-scale features is superior in image classification.

Keywords: multi-scale features; visual attention mechanism; convolutional neural network; image classification

1. Introduction

In recent years, deep learning has made a number of breakthroughs in the fields of computer vision [1,2], natural language processing [3,4], and speech recognition [5,6]. As one of the most typical deep learning models, convolutional neural networks (CNNs) have made considerable progress in image classification [7,8], object detection [9,10], image retrieval [11,12], and other applications. With the richness of image datasets and the improvement of machine performance, CNN's powerful feature extraction and generalization capabilities are increasingly favored by the industry. Several typical CNN models (including AlexNet [13], VGG [14], ResNet [15], etc.) were originally used for image classification and further demonstrated its versatility in other image processing tasks. This article will discuss a different convolution model and apply it to image classification.

The visual attention mechanism has proven to be one of the most fascinating areas of cognitive neuroscience research. Human vision can quickly scan global input information and screen out specific targets. That is to say, it has the ability to pay attention to certain things while ignoring other things. By mimicking human visual attention, early attention models are often divided into data-driven

bottom-up models [16–18] and task-driven top-down models [19–21]. The research on the combination of deep learning and attention mechanisms has received considerable attention [22–24]. The goal of this study was to select information from the input that was relatively important to the current task. In deep neural networks, researchers often use masks to achieve attention. They demarcate key features in the data by training additional multi-layer weights. This approach naturally embeds the attention mechanism into the deep network structure and participates in end-to-end training. Attention models are well suited for solving computer vision tasks such as image classification, saliency analysis, and object detection.

The same object will show different shapes in different natural scenes. When a computer vision system senses an unfamiliar scene, it cannot predict the scale of the object in the image in advance. Therefore, it is necessary to observe image information at different scales. The multi-scale representation of images can be divided into two types: Multi-scale space and multi-resolution pyramid. The difference between them is that multi-scale space has the same resolution at diverse scales. In the visual tasks, the multi-scale approach of multi-resolution pyramid processing targets can be separated into two categories: Image pyramid and feature pyramid. The image pyramid works best but the time and space complexity is high, and the feature pyramid is widely used because of its ease of implementation. In addition, the sizes of the receptive field and the number of network layers of CNN can also obtain multi-scale features in nature. The smaller receptive field and shallower network layers can focus on local features, the larger receptive field and deeper network layers can perceive global information.

CNN has a natural advantage for image processing. At the same time, the attention mechanisms and multi-scale features are effective means of image processing. Therefore, this paper combines attention mechanisms and multi-scale features to create a new convolution model for image classification. Res2Net [25] groups single residual block and constructs hierarchical residual connection between different groups to achieve multi-scale representation of features. However, the randomness of its grouping method affects the robustness and intuitive interpretability of the network. We first sort channels of a single residual block by the attention mechanism, and then group the sorted residual blocks. This can achieve multi-scale representation of features, highlight important features and suppress interference information. Further, we conduct an attentive analysis for each group to improve the accuracy of image classification.

2. Related Work

2.1. Different Convolution Methods

Convolution is the core operation of CNN for feature extraction. LeNet [26] first used the "single-layer convolution + single-layer pooling" structure. In order to extract more features, AlexNet [13], VGG [14], etc., began to adopt a "multi-layer convolution + single-layer pooling" structure. These network structures only improve the way in which the convolutions are superimposed. NIN [27] employs a multi-layer perceptron convolutions instead of traditional convolutions, where "1*1" convolution can adjust the number of hidden layers and increase the nonlinear expression of the network. Inception [28–30] series performs convolution operations in a single convolutional layer with multiple different receptive fields, which take into account multi-scale feature learning of the network. Depth-wise separable [31,32] convolution considers the channel and space of the feature map separately by the depth-wise convolution process and the point-wise convolution process. ShuffleNet [33] and IGCNets [34] increase the interaction between different groups through different forms of grouping and interleaved convolution. These network models have improved the sizes of the convolution kernels, including the widths, heights, and number of channels. Dilated convolution [35] multiplies the range of receptive fields by inserting "0" into the convolution kernel. Deformable convolution [36] gives the convolution kernel deformation capability according to given rules. SPS-ConV [23] uses the improved SLIC algorithm for super-pixel segmentation of images. This cluster-based convolution is

easier to adapt to the complex geometric transformation of the image. These convolutional network models have improved the form of the convolution kernel.

Here are some singular convolution models. Selective Kernel networks [37] allows the network to adaptively adjust the receptive field sizes based on the input multi-scale information. Highway networks [38] allow unimpeded information flow across several layers on information highways. WideResNet [39] is an uncharacteristic example. It increases the performance of the network by expanding the width of the network instead of the depth. The feature recalibration convolution [22] automatically acquires the importance of each feature channel through self-learning, and then enhances useful features and suppresses unhelpful features according to their importance. Res2Net [25] proposes a new convolution backbone that groups single residual blocks and integrates more granular convolution operations. We try to give this convolution backbone a whole new meaning. First, the importance of each channel in a residual block is sorted by feature recalibration convolution. Then, the sorted residual blocks are grouped and layer-wise convolutions are performed. Finally, the results of the layer-wise convolution are concatenated and channel fusion is performed using a "1 × 1" convolution.

2.2. Multi-Scale Features

Here we only discuss multi-scale representation of images based on multi-resolution pyramids, which can be divided into two categories: Image pyramid and feature pyramid. Classic pedestrian detection algorithms, such as "Haar + Adaboost" and "HOG + SVM", use image pyramids to handle multi-scale targets. The image pyramid is a multi-scale representation of a single image. It consists of a series of different resolution versions of the original image. The Gaussian pyramid and the Laplacian pyramid are two common image pyramids, the former for downsampling images and the latter for upsampling images. Image pyramid processing multi-scale representation can achieve optimal results, but its time and space complexities are too high. Therefore, the feature pyramid becomes the protagonist of multi-scale representation.

U-Net [40] adopts a symmetric encoder–decoder structure to combine high-level features with low-level features to obtain richer multi-scale information. SSD [41] uses feature maps from different convolutional layers for object detection at different scales. FPN [42] employs the feature pyramid to scale the features of different layers, and then performs information fusion. Built on the FPN, PANet [43] adds a bottom-up pyramid to effectively transfer positioning information to other layers. ZigZagNet [44] further improved PANet, allowing different layers to interact to enhance bidirectional multi-scale context information. MSPN [45] and M2Det [46] have each completed two or even multiple FPN cascades, which proves that a simple superimposed FPN structure can obtain multi-scale features more effectively. UPerNet [47] and Parsing R-CNN [48], respectively, concatenate PPM modules and RoIAlign operations with FPN to obtain rich multi-scale information.

Most feature pyramid methods represent multi-scale features in a layer-wise manner. Res2Net [25] learns multi-scale representations from a novel and more granular perspective. It makes a multi-scale representation into each residual block, allowing a single network layer to have different sizes of receptive fields. However, Res2Net's grouping of residual blocks is random. We attempted to introduce attention mechanisms into the grouping tasks of Res2Net, so that more important information has the most hierarchical feature representation.

3. Approach

3.1. Multi-Scale Convolutional Model Based on Single Attention

ResNet proved that using residual blocks to learning the residuals between inputs and outputs is simpler and more efficient than learning the mapping between them directly. Res2Net further adopts grouping convolution to construct finer quadratic residual connections in the residual block, allowing the network to have richer receptive fields to learn multi-scale features. However, the random grouping

of Res2Net in the residual block affects the robustness and intuitive interpretability of deep neural networks. This section will introduce a multi-scale convolutional model based on single attention (AMS-CNN). AMS-CNN is superimposed by multiple AMS-blocks. Its core content is the addition of the "Channels-Sort" module to Res2-block. The "Channels-Sort" module ranks each channel in the residual block in descending order by feature importance. The AMS-block then groups the sorted residual blocks and performs secondary residual learning. Finally, the "1 × 1" convolution is used to adjust the number of channels and fuse all channel features. AMS-bclok subtly combines the attention mechanism with multi-scale features through the Channels-Sort module, which can focus on specific targets and achieve multi-scale observation of objects. The AMS-block structure is shown in Figure 1.

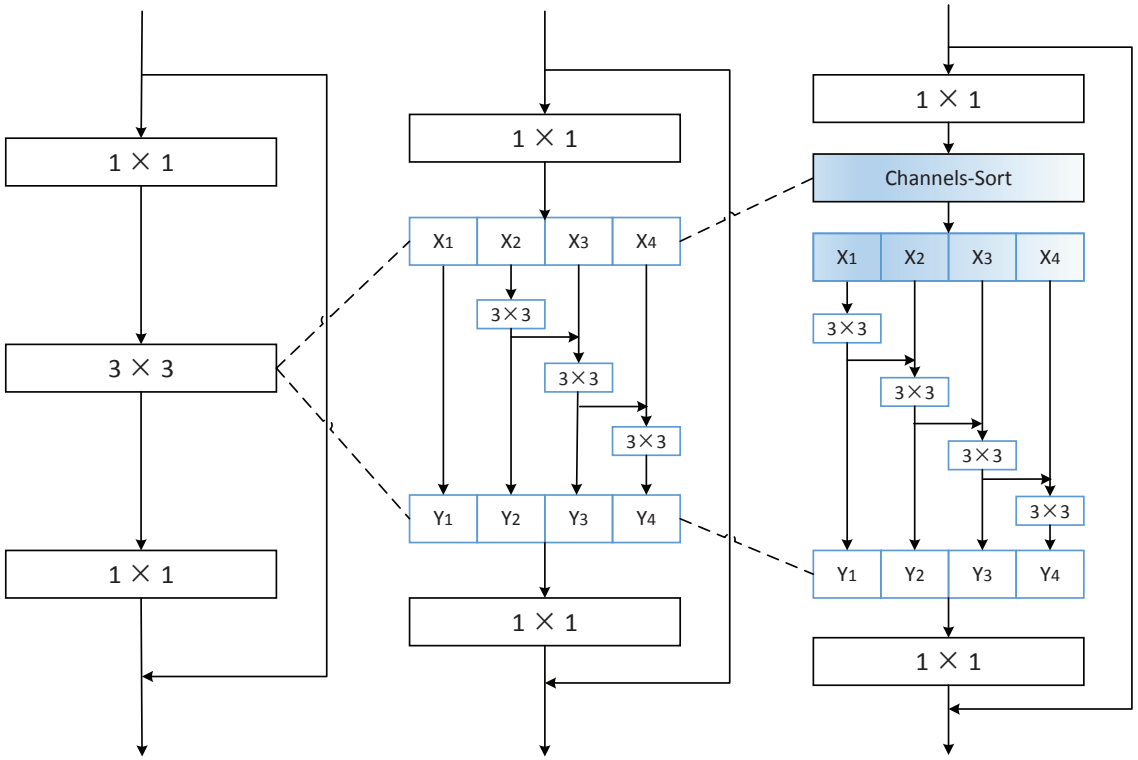


Figure 1. The structure schematic of three blocks. The former is Res-block, the middle represents Res2-block, and the latter describes AMS-block.

As shown in Figure 1, Res2-block is a decomposition of the traditional Residual block, so that the model can learn multi-scale features in a single residual block. AMS-block introduces the Channels-Sort module based on the structure of Res2-block. The other difference from Res2-block is that we perform secondary residual learning for each grouping of the residual block, as shown in the right of Figure 1, where X_1 is transformed to Y_1. This can further enrich the receptive fields of convolution operations and increase multi-scale information. The structure of the Channels-Sort module is illustrated in Figure 2.

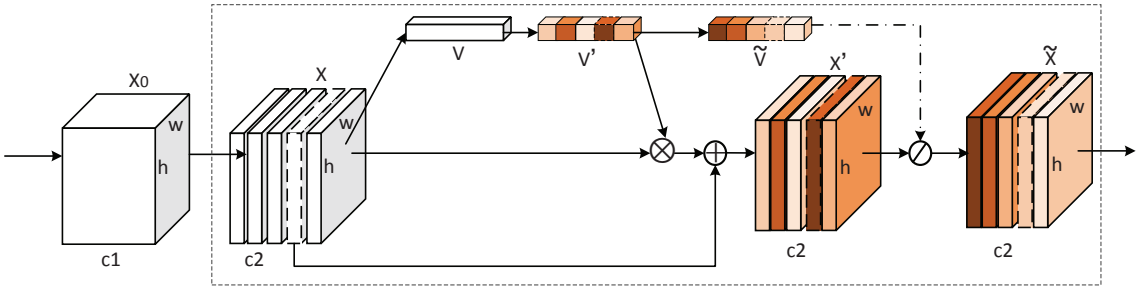


Figure 2. The structure schematic of Channels-Sort module.

As shown in Figure 2, X_0 obtains X via a normal convolution operation or a convolution operation that introduces channel attention [22,24] or space attention [23,24]. The dotted box indicates the Channels-Sort module. The first half performs the Squeeze-and-Excitation operation, and the second half sorts the channels of the feature maps in descending order according to the feature importance learned by Squeeze-and-Excitation. The process of transforming X into V adopts two methods: Global average-pooling and depth-wise convolution, where $V = [V1, V2]$.

$$V_1 = DWConV(K_a, X) = \sum_{i=1}^{c2}(K_{ai} \odot X_i) \tag{1}$$

$$V_2 = GlobleAvgPool(X) \tag{2}$$

$$V = ReLU(Concat(V_1, V_2)) \tag{3}$$

where "$DWConV$" means depth-wish convolution [31], and K_a is the convolution kernel of X to V, K_a = [K_{a1}, K_{a2}, ..., K_{ac2}], where the number of channels of the convolution kernel is 1. "$GlobleAvgPool$" represent global average-pooling operation. Finally, V_1, and V_2 are connected in series along the direction of the feature channel and activated by the $ReLU$ function to obtain V.

$$V^{'} = Sigmoid(ConV(K_b, V)) = Sigmoid(K_{b1} \odot V) \tag{4}$$

After the convolution and activation operations, V obtains the vector V' that characterizes the importance of the feature channels. The convolution kernel is K_b (K_b = [K_{b1}]) and the activation function is $ReLU$.

$$X^{'} = Add(Multiply(V^{'}, X), X) = (V^{'} \otimes X) \oplus X \tag{5}$$

where V' represents an important factor of the feature channel. X' is first multiplied by X and V' element by element, and then the result is added to X element by element. The $\otimes$ and the $\oplus$ represent element-by-element multiplication and addition, respectively.

$$\widetilde{V} = Top\_K(V^{'}, c2) \tag{6}$$

$$\widetilde{X} = Batch\_Gather(X^{'}, \widetilde{V}) \tag{7}$$

The function Top_K(data,k) indicates that the first k data are extracted in order. As Equation (6), when the value of k is the channel number $c2$, it means that we sort the vector V' in descending order. The $\oslash$ in Figure 2 represents channels sorting of X' based on the index values of $\widetilde{V}$, which can be implemented by the Tensorflow built-in function $Batch\_Gather(*)$. The implementation algorithm of AMS-block is as shown in Algorithm A1 (Appendix A).

3.2. Multi-Scale Convolutional Model Based on Multiple Attention

The residual block with feature importance ordering processed by the Channels-Sort module is divided into n small blocks by equal grouping operation, where $X = [X_1, X_2, ..., X_k, ..., X_n]$. From front to back, the importance of each group to the classification results is gradually reduced. Therefore, it enables more important features to expresses in more scales. Next, we introduce a spatial attention block before the Channels-Sort module to capture more distinguished local features, forming a multi-scale convolutional block with dual attention (DAMS-block). This process can be obtained in the X to Y stages of Figure 2. For a more fine-grained image classification, we further constructed the multi-scale convolutional block with multiple attention (MAMS-block). On the basis of DAMS-bclok, we pay attention to each residual small block after grouping. The channel attention analysis is

performed on the small blocks in the same residual block in order to better realize the feature fusion between different small blocks. A network structure in which a plurality of MAMS-blocks is superimposed is called MAMS-CNN. The DAMS-block and the MAMS-block are given in Figure 3.

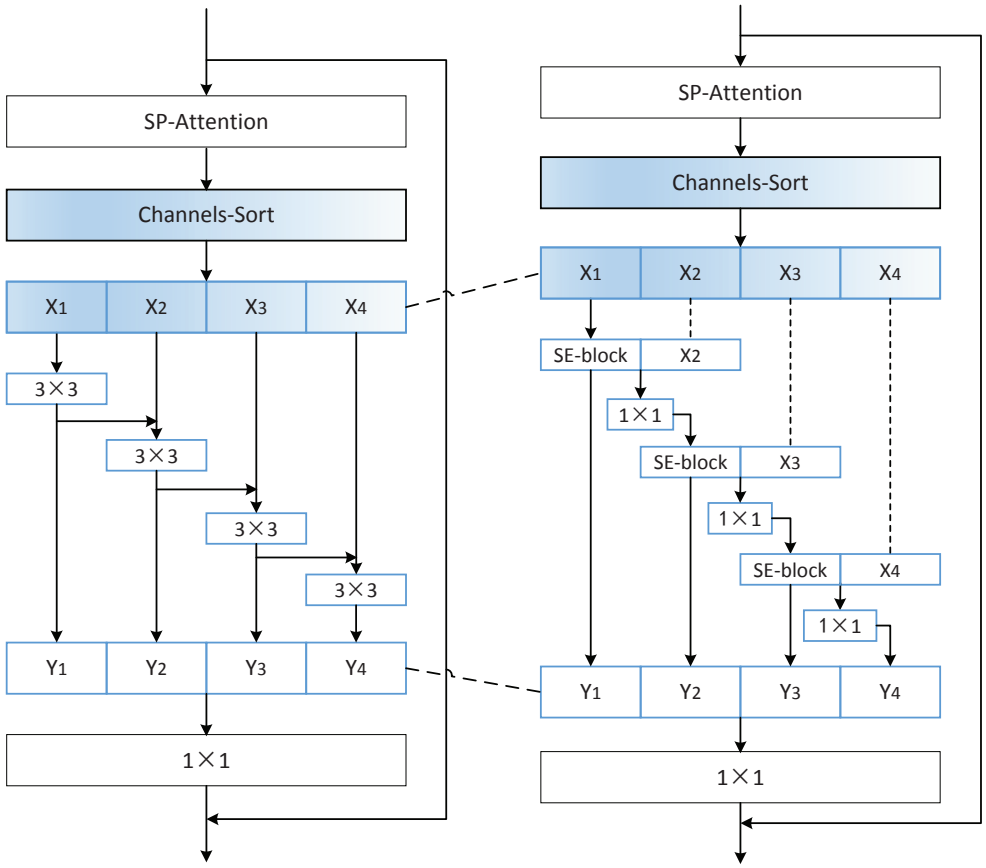


Figure 3. The structure schematic of DAMS-block and MAMS-block.

As shown in Figure 3, channel attention analysis is performed in each residual small block, and more detailed features can be observed in the receptive fields of different scales. For the operation between the small blocks, we chose to use the channel fusion of "1 × 1" convolution instead of adding them by element. The main reason is that different small groups focus on different features. Simple addition will blur the attention information, but channel fusion will get more attention features. Specific operations are as shown in Equations (8) and (9).

$$F_{se}: \begin{cases} F_{sq}: V = GlobleAvgPool(X_k) \\ F_{ex}: V^{'} = Sigmoid(Dense(Dense(V, c/r), c)) \\ F_{scale}: X = Add(Multiply(V^{'}, X_k), X_k) \end{cases} \tag{8}$$

$$Y_k = \begin{cases} F_{se}(X_1) & k = 1 \\ F_{se}(ConV(Concat(Y_{k-1}, X_k), \{1 \times 1\})) & 1 < k < n \\ ConV(Concat(Y_{n-1}, X_n), \{1 \times 1\}) & k = n \end{cases} \tag{9}$$

The "SE-block" in Figure 3 performs the $F_{se}(*)$ operation. Where $F_{se}(*)$ means to perform a Squeeze-and-Excitation operation, that is, channel attention. $F_{se}(*)$ is divided into three phases: Squeeze operation, excitation operation, and reweight operation. See reference [22] for details.

The parameter n represents the number of groups per residual block (e.g., n = 4 in Figure 3). $Conv(*,\{1 \times 1\})$ fuses channel information and adjusts the channel number using "1×1" convolution. $Concat(*)$ means that two convolutional blocks are connected in series along the direction of the feature channel.

4. Experiments

4.1. Datasets and Experimental Details

In this section, we perform experiments on four image datasets, including CIFAR-10, CIFAR-100, FGVC-Aircraft [49], and Stanford Cars [50]. The CIFAR-10 consists of 60,000 32×32 images in 10 classes, with 6000 images per class. There are 50,000 training images and 10,000 test images. The CIFAR-100 is just like the Cifar-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The FGVC-Aircraft consists of 10,000 images. They are divided into 100 categories. This dataset includes 6667 training images and 3333 test images. The Stanford Cars consists of 16,185 images in 196 categories. This dataset is splited into 8144 training images and 8041 test images. Figure 4 shows partial examples of these datasets.

(**a**) CIFAR-10 (**b**) FGVC-Aircraft (**c**) Stanford Cars

Figure 4. The schematic images of four datasets.

The algorithm in this manuscript is implemented on the Keras 2.2.4 framework backed by Tensorflow 1.11.0. In addition, the graphics card model is a GeForce GTX 1080 and the programming language and version is Python 3.5.2. The first half of epoches has a learning rate of 0.1, the third quarter of epoches has a learning rate of 0.01, and the last quarter of epoches has a learning rate of 0.001. For example, when epoches are equal to 300, the learning rates of the models were 0.1, 0.01, and 0.001, at 1–150 epochs, 151–225 epochs, and 226–300 epochs. The experiments adopt the SGD algorithm with momentum, where the momentum value is 0.9. Convolutional network structures, such as ResNet-50 and ResNet-101, are not used, mainly because their parameters are too large. The improved models based on these mainstream convolutional structures are difficult to complete under our existing experimental conditions. We only verify the validity and feasibility of the proposed methods with convolution models of the same magnitude.

4.2. Experiments on CIFAR-10 and CIFAR-100

In this section, we adopt ResNet, SE-CNN and Res2-CNN as reference to verify the effectiveness of our proposed models from three aspects: Test accuracy, time complexity, and computational complexity. Among them, SE-CNN and Res2-CNN, respectively, add SE-blocks and Res2-blocks to the ResNet-32 structure. The structures of ResNet-32 and its various improved models are shown in Table 1. ResNet-32 consists of four convolutional blocks, a global average pooling layer, and a dense layer. There are two convolution operations in one residual block, in each square bracket. The three parameters of the convolution kernel are the number of feature channels after convolution, the sizes of the convolution kernels, and the strides. When dimension reduction is required, the strides are

equal to 2. The number of repetitions of each residual block is established according to the subsequent parameter "*stack_n*", which is 5, 4, or 1 in ResNet-32.

Table 1. The simple structures of several convolutional networks with residual module.

Model	Conv1	Conv2	Conv3	Conv4	Dence
ResNet-32	(n1, 3 × 3, 1)	$\begin{bmatrix} \text{n1}, 3\times3, 1 \\ \underline{\boldsymbol{n2,\ 3\times3,\ 1}} \end{bmatrix} \times 5$ $\begin{bmatrix} \text{n2}, 3\times3, 2 \\ \underline{\boldsymbol{n2,\ 3\times3,\ 1}} \end{bmatrix} \times 1$	$\begin{bmatrix} \text{n2}, 3\times3, 1 \\ \underline{\boldsymbol{n3,\ 3\times3,\ 1}} \end{bmatrix} \times 4$ $\begin{bmatrix} \text{n3}, 3\times3, 2 \\ \underline{\boldsymbol{n3,\ 3\times3,\ 1}} \end{bmatrix} \times 1$	$\begin{bmatrix} \text{n3}, 3\times3, 1 \\ \underline{\boldsymbol{n3,\ 3\times3,\ 1}} \end{bmatrix} \times 4$	GAP (8,8) Dence (10/100)
SE-CNN		–	–		
Res2-CNN_A/B		Res2-block	Res2-block		
AMS-CNN_A/B		AMS-block	AMS-block		
DAMS-CNN_A/B		DAMS-block	DAMS-block		
MAMS-CNN_A/B		MAMS-block	MAMS-block		

As showed in Table 1, for CIFAR-10 and CIFAR-100, the number of convolution channels $n1$, $n2$, and $n3$ of ResNet-32 takes values of 16, 32, 64, and 32, 64, 128, respectively. Several other network models are implemented on the basis of ResNet-32. SE-CNN adds SE-block to each residual block of ResNet-32, adding a total of 15 SE-blocks. Considering the computational complexity and time complexity, we designed both *A* and *B* structures for the proposed methods. As showed in Table 1, Structure *A* adds a corresponding block after *Conv*2 and *Conv*3 of ResNet-32. Structure *B* replaces the underlined convolutional layers with the corresponding block. The corresponding blocks include AMS-block, DAMS-block and MAMS-block. For comparison purpose, Res2-CNN uses the same two architectures, such as Res2-CNN_A and Res2-CNN_B.

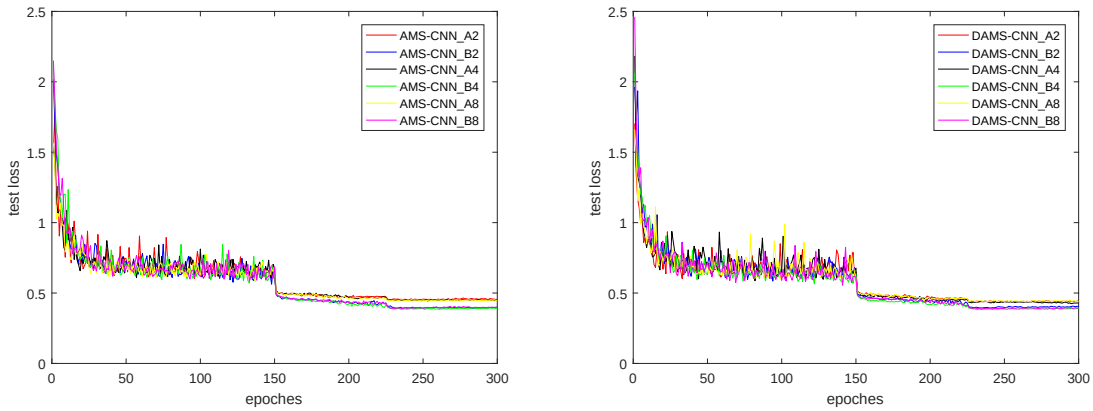
Table 2 shows the test accuracy and time complexity of CIFAR-10 under different models. The parameter "*Groups*" indicate the number of grouping feature channels. The values in the experiments are 2, 4, and 8. Here ResNet and SE-CNN do not perform group convolution, which is commensurate with a value of 1. The parameter "*SPE*" means "seconds per epoch", which is the number of seconds required to execute each epoch. Owing to the different grouping methods, the parameters that need to be trained throughout the networks are also very different. For comparison, ResNet and SE-CNN adopt a 32-layer network structure, and other models adopt distinct layers to make the overall parameters quantity close to ResNet-32.

For ResNet-32, the classification accuracy of CIFAR-10 is as high as 92.49% with only $0.46M$ parameter. SE-CNN introduces SE-block in the ResNet-32 architecture, which increased the test accuracy by 0.26%. Res2-CNN_A and Res2-CNN_B have improved the test accuracy by 0.28% and 0.39%, respectively, and their time complexity has gradually increased. Comparing different grouping forms of Res2-CNN, it can be concluded that the best result is achieved when the number of groups is 4. It can be seen from Table 2 that under the condition of only adding a small number of parameters, the classification accuracy of our proposed model is over 93%. Compared to ResNet, the test accuracy classification of AMS-CNN_A and AMS-CNN_B increased by 0.82% and 0.86%. Unlike Res2-CNN, the more channel groupings of AMS-CNN has, the better its performance, which mainly reflects the advantages of multi-scale functions. DAMS-CNN added spatial attention to AMS-CNN. The test accuracy classification of the two structures increased by 1.09% and 1.11%, and the highest result was 93.60%. Overall, the time complexity of several models corresponding to structure A is lower, while structure B takes more time. The main time consumption is in the process of sorting the channels by feature's importance. As can be seen from the FLOPs values, the computational complexities of the algorithms are on the same order of magnitude. This is because we control the size of the parameters of the network. Figure 5 shows the trend of changes in the test loss of CIFAR-10 for various network models. The downward tendency of structure B is more obvious than that of structure A, indicating

that the more AMS-block or DAMS-block is used, the more powerful the network is. The overall performance of DAMS-CNN is better than AMS-CNN.

Table 2. The FLOPs (floating point operations) and SPE (seconds per epoch) of the residual networks carrying different convolution models, and the classification accuracies of these networks on the CIFAR-10 dataset.

Model	Groups	Accuracy (%)	Parameters	SPEFPS	FLOPs
ResNet-32	1	92.49	0.46M	22	1.87×10^6
SE-CNN-32	1	92.75	0.47M	29	1.90×10^6
	2	92.64	0.49M	26	1.96×10^6
Res2-CNN_A	4	92.77	0.49M	27	1.95×10^6
	8	92.25	0.48M	29	1.94×10^6
	2	92.77	0.49M	55	1.97×10^6
Res2-CNN_B	4	92.88	0.47M	65	1.88×10^6
	8	92.82	0.44M	90	1.78×10^6
	2	93.16	0.50M	29	2.01×10^6
AMS-CNN_A	4	93.13	0.49M	33	1.97×10^6
	8	93.31	0.49M	37	1.95×10^6
	2	93.18	0.50M	80	2.01×10^6
AMS-CNN_B	4	93.14	0.50M	112	1.72×10^6
	8	93.35	0.46M	169	1.57×10^6
	2	93.58	0.50M	31	2.02×10^6
DAMS-CNN_A	4	93.39	0.49M	33	1.97×10^6
	8	93.41	0.49M	37	1.95×10^6
	2	93.33	0.50M	91	2.01×10^6
DAMS-CNN_B	4	93.39	0.50M	128	1.72×10^6
	8	93.60	0.46M	185	1.58×10^6



(**a**) AMS-CNN (**b**) DAMS-CNN

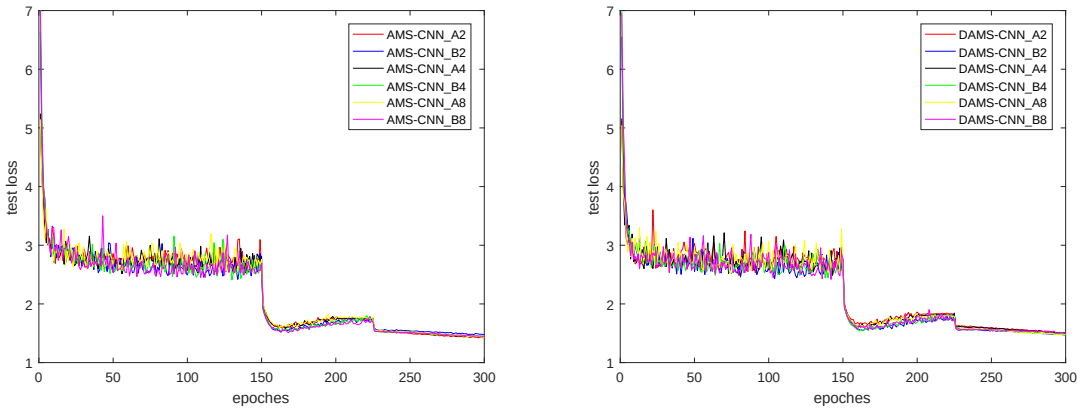
Figure 5. Test loss of various network structures on CIFAR-10.

We adopt ResNet-32 as the network foundation model and limit the network parameters around 1.9M. Table 3 shows the test accuracies and time consumption of ResNet-32 and its various deformation models on the CIFAR-100. As before, we add an SE-block to each residual block of ResNet-32 to form SE-CNN. In addition, Res2-CNN, AMS-CNN and DAMS-CNN employ both A and B structures. At the same time, we have different groupings for the proposed models. We did not use the MAMS-CNN model for the CIFAR-10 and CIFAR-100 datasets. This is because the CIFAR datasets is relatively simple, and an overly complex feature learning model can lead to overfitting. It is worth noting that when we train AMS-CNN and DAMS-CNN, we increase the network weight attenuation coefficient "*weight_decay*" from 0.0001 to 0.0005, in order to avoid over-fitting caused by complex models.

Table 3. The FLOPs and SPE (seconds per epoch) of the residual networks carrying different convolution models, and the classification accuracies of these networks on the CIFAR-100 dataset.

Model	Groups	Accuracy (%)	Parameters	SPE	FLOPs
ResNet-32	1	73.09	1.87*M*	32	7.50×10^6
SE-CNN	1	74.19	1.90*M*	42	7.61×10^6
	2	73.47	1.96*M*	36	7.85×10^6
Res2-CNN_A	4	73.48	1.95*M*	37	7.80×10^6
	8	74.19	1.94*M*	39	7.75×10^6
	2	74.35	1.95*M*	80	7.81×10^6
Res2-CNN_B	4	74.35	1.86*M*	94	7.47×10^6
	8	75.17	1.76*M*	117	7.05×10^6
	2	75.32	2.01*M*	40	8.06×10^6
AMS-CNN_A	4	75.25	1.97*M*	42	7.88×10^6
	8	75.36	1.95*M*	47	7.78×10^6
	2	75.31	1.98*M*	111	7.96×10^6
AMS-CNN_B	4	75.87	1.98*M*	154	6.80×10^6
	8	75.42	1.82*M*	201	6.22×10^6
	2	75.52	1.99*M*	38	7.99×10^6
DAMS-CNN_A	4	75.34	1.97*M*	43	7.80×10^6
	8	75.25	1.94*M*	48	7.71×10^6
	2	75.41	1.99*M*	125	7.97×10^6
DAMS-CNN_B	4	75.38	1.99*M*	167	6.81×10^6
	8	75.43	1.82*M*	216	6.23×10^6

As can be seen from Table 3, when using *mode_A* training for various networks, the number of seconds it takes to process the images of the entire dataset is almost the same. However, when using the *mode_B*, the time complexity will increase a lot. For example, the *SPE* value of DAMS-CNN_B is more than six times that of ResNet-32. Under the same conditions, the performance of the three models we proposed exceeding the capabilities of Res2-CNN, SE-CNN, and ResNet. For the CIFAR-100, we achieved a maximum test accuracy of 75.87%, which was 2.78%, 1.68%, and 2.39% higher than the previous three comparison models. In addition, the performance of AMS-CNN_B is higher than that of AMS-CNN_A, which shows that adopting more AMS-block is beneficial to the improvement of network performance. Moreover, the performance of DAMS-CNN_A is better than that of AMS-CNN_A, indicating that the introduction of spatial attention mechanism on the basis of the latter is beneficial to feature learning. However, the overall performance of DAMS-CNN_B is not higher than DAMS-CNN_A. Through analysis, we believe that the network has been over-fitted. Therefore, we have not discussed the more complex model (MAMS-CNN) for the basic CIFAR dataset. Similar to the previous experiments, the computational complexities of the algorithms are on the same order of magnitude. Similar to Figure 5, structure B in Figure 6 is more efficient than the structure A, while DAMS-CNN is better than AMS-CNN. The overall loss of CIFAR-100 is large and fluctuates greatly during training. This happens because the image classification of the FIFAR-100 is more complex.



(**a**) AMS-CNN (**b**) DAMS-CNN

Figure 6. Test loss of various network structures on CIFAR-100.

4.3. Experiments on Fine-Grained Image Datasets

In this section, we will adopt the proposed models for fine-grained image classification, including FGVC-Aircraft and Stanford Cars. Fine-grained images have the character of "small differences between classes and large differences within classes". Compared to the CIFAR dataset, we experimented with more complex network structures. All of the models in Table 4 are based on the ResNet structure. The ResNet structure here includes six convolution blocks, a global average pooling layer, and a dense layer. The number of output nodes of the networks is 100 and 196 depending on the datasets. The three parameters of each convolutional layer represent the number of channels after convolution, the size of the convolution kernel, and the stride. The default value of "*Stride*" is 1, which can be omitted. "*Conv*1" uses three different scales of receptive fields to capture richer features. The convolution kernels in all remaining residual blocks are 3×3, 3×1, and 1×3, in order to obtain rich features while reducing parameters. SE-CNN adds an SE-block to each residual block in ResNet for a total of 21 SE-blocks. For Res2-CNN, AMS-CNN, DAMS-CNN, and MAMS-CNN, we only experimented with the structure A mentioned in the previous section. This is to reduce the complexity of the models in order to complete the experiments under limited equipment conditions. We add Res2-block, AMS-block, DAMS-block, and MAMS-block after each large convolution block. Four corresponding special modules are added to each convolution model.

Next, we will compare the classification accuracy and time complexity of the six convolution models of ResNet, SE-CNN, Res2-CNN, AMS-CNN, DAMS-CNN, and MAMS-CNN for fine-grained image sets under the same volume parameters. The trainable parameters of the networks in the experiments are between $10M$ and $11M$. Among them, four models (Res2-CNN, AMS-CNN, DAMS-CNN, and MAMS-CNN) employ group convolution and the number of channel groupings are unified to 4. Table 5 gives the experimental results for the FGVC-Aircraft dataset; Table 6 shows the experimental data for the Stanford Cars dataset. In the experiment, we compressed the size of the images of the two datasets into 224×224. Due to the limitations of the experimental equipment, we did not choose to use the commonly used image size of 448×448. The learning rates of the models were 0.1, 0.01, and 0.001, at 1–190 epochs, 191–270 epochs, and 271–360 epochs. The network weight attenuation coefficient "*weight_decay*" is set to 0.0005, in order to avoid over-fitting caused by complex models. It is worth noting that we did not use pre-training networks and fine-tuning techniques throughout the experiments.

Table 4. The simple structures of several convolutional networks with residual module.

Model	Conv1	Conv2	Conv3	Conv4	Conv5	Conv6	Dence
ResNet	(16, 7, 2) (16, 5, 2) (16, 3, 2)	$\begin{bmatrix} 48, 3\times 3 \\ 48, 3\times 1 \\ 48, 1\times 3 \end{bmatrix}\times 3$ $\begin{bmatrix} 64, 3\times 3, 2 \\ 64, 3\times 1, 1 \\ 64, 1\times 3, 1 \end{bmatrix}$	$\begin{bmatrix} 64, 3\times 3 \\ 64, 3\times 1 \\ 64, 1\times 3 \end{bmatrix}\times 2$ $\begin{bmatrix} 128, 3\times 3, 2 \\ 128, 3\times 1, 1 \\ 128, 1\times 3, 1 \end{bmatrix}$	$\begin{bmatrix} 128, 3\times 3 \\ 128, 3\times 1 \\ 128, 1\times 3 \end{bmatrix}\times 2$ $\begin{bmatrix} 192, 3\times 3, 2 \\ 192, 3\times 1, 1 \\ 192, 1\times 3, 1 \end{bmatrix}$	$\begin{bmatrix} 192, 3\times 3 \\ 192, 3\times 1 \\ 192, 1\times 3 \end{bmatrix}\times 5$ $\begin{bmatrix} 256, 3\times 3, 2 \\ 256, 3\times 1, 1 \\ 256, 1\times 3, 1 \end{bmatrix}$	$\begin{bmatrix} 256, 3\times 3 \\ 256, 3\times 1 \\ 256, 1\times 3 \end{bmatrix}\times 5$	GAP (7, 7) Dence (100/196)
SE-CNN		–	–	–	–		
Res2-CNN		Res2-block	Res2-block	Res2-block	Res2-block		
AMS-CNN		AMS-block	AMS-block	AMS-block	AMS-block		
DAMS-CNN		DAMS-block	DAMS-block	DAMS-block	DAMS-block		
MAMS-CNN		MAMS-block	MAMS-block	MAMS-block	MAMS-block		

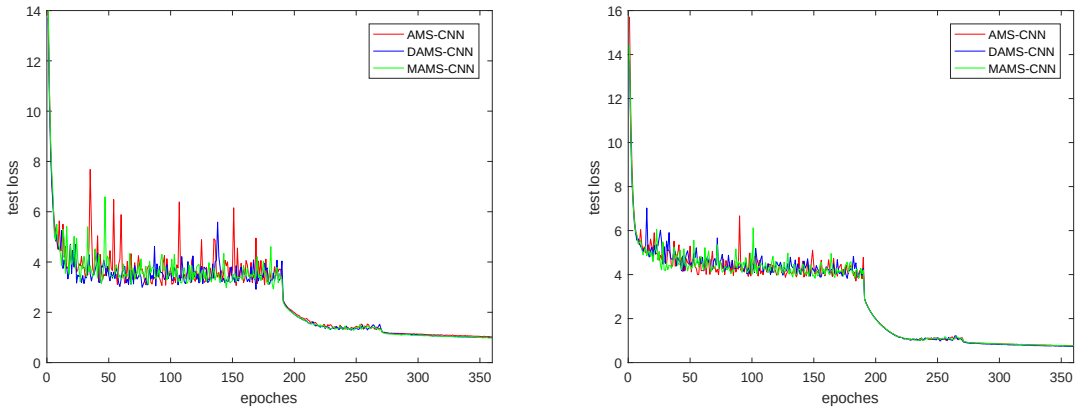
Table 5. The FLOPs and SPE (seconds per epoch) of the residual networks carrying different convolution models, and the classification accuracies of these networks on the FGVC-Aircraft dataset.

Model	Accuracy (%)	Parameters	SPE	FLOPs
ResNet	83.05	10.32M	87	4.13×10^7
SE-CNN	84.22	10.39M	95	4.17×10^7
Res2-CNN	82.27	10.71M	94	4.35×10^7
AMS-CNN	85.42	10.85M	100	4.35×10^7
DAMS-CNN	85.57	10.85M	104	4.58×10^7
MAMS-CNN	86.56	10.90M	111	4.37×10^7

Table 6. The FLOPs and SPE (seconds per epoch) of the residual networks carrying different convolution models, and the classification accuracies of these networks on the Stanford Cars dataset.

Model	Accuracy (%)	Parameters	SPE	FLOPs
ResNet	82.84	10.33M	119	4.12×10^7
SE-CNN	83.09	10.41M	132	4.16×10^7
Res2-CNN	83.80	10.73M	129	4.28×10^7
AMS-CNN	88.65	10.87M	142	4.34×10^7
DAMS-CNN	89.02	10.87M	146	4.34×10^7
MAMS-CNN	89.15	10.93M	156	4.36×10^7

The parameters of the networks are shown in Tables 5 and 6, and the values are between 10M and 11M. Since the datasets FGVC-Aircraft and Stanford Cars, respectively, contain images with 100 and 196 categories, the final output nodes of the networks are different. Therefore, the amount of parameters in Table 6 is slightly higher overall than Table 5. For FGVC-Aircraft, the test accuracies of MAMS-CNN reached 86.56%, which was 3.51%, 2.34%, and 4.29% higher than ResNet, SE-CNN, and Res2-CNN, respectively. At the same time, the performance of multi-scale networks with single attention, dual attention and multiple attention is getting higher and higher, indicating that the attention mechanisms have a natural advantage in image classification. It is worth noting that Res2-CNN does not show good performance on FGVC-Aircraft. For Stanford Cars, the MAMS-CNN network structure also achieved the best results, with a test accuracy rate of 89.15%, which is 6.13%, 6.06%, and 5.53% higher than ResNet, SE-CNN, and Res2-CNN, respectively. Even compared to the AMS-CNN and DAMS-CNN models, their performance is also improved by 0.50% and 0.13%, respectively. On the other hand, it can be seen that the more complicated the network structure is, the more time is consumed. However, for a dataset with an overall number of more than 10,000 images, the time complexity is within the tolerable range. Comparing two fine-grained datasets, we can see that our proposed models have better improvement effect on Stanford Cars than FGVC-Aircraft, because the former has more local features for fine-grained image recognition than the latter. As can be seen from the FLOPs and SPE, under similar computational complexity conditions, the time complexities are significantly different because the Channels-Sorting consumes a lot of time. As can be seen in Figure 7, from AMS-CNN to MAMS-CNN, the performance of the test loss is more stable. Comparing the three structures on two different datasets, it can be seen that the test loss on Stanford Cars are much less fluctuating and the final loss values are smaller. This is consistent with the performance of the networks on both datasets.



(**a**) FGVC-Aircraft (**b**) Stanford Cars

Figure 7. The test loss of various network structures on FGVC-Aircraft and Stanford Cars.

4.4. Comparisons with Prior Methods

In the previous sections, our focus was to demonstrate the effectiveness of the proposed modules on the same scale. we have added comparative experiments for the four datasets. We have added comparative experiments for the CIFAR-10 and CIFAR-100. In addition we compare our results with several other fine-grained classification models, including FV-CNN, DVAN, Random Maclaurin, Tensor Sketch, LRBP. All of these models do not use bounding box information and part annotation. The classification accuracies of the mentioned methods are shown in Tables 7 and 8.

Table 7. The classification accuracies of these models on the CIFAR datasets.

Model	Accuracy (%)	
	CIFAR-10	**CIFAR-100**
NIN [27]	91.19	64.33
HighWay Net [38]	92.28	67.61
ResNet-110 [15]	93.57	78.84
SENet [22]	95.60	–
WideResNet(28, 10) [39]	95.35	79.87
MAMS-CNN (Ours)	95.83	80.03

Table 8. The classification accuracies of different models on the two fine-grained datasets.

Model	Accuracy(%)	
	Aircrafts	**Cars**
FV-CNN [51]	81.46	87.79
DVAN [52]	–	87.10
Random Maclaurin [53]	87.10	89.54
Tensor Sketch [53]	87.18	90.19
LRBP [54]	87.31	90.92
MAMS-CNN (Ours)	87.72	90.89

For the CIFAR dataset, the structure of our model is built on the WideResNet structure, adding three MAMS-CNN modules. For the fairness of the experiment, we ran WideResNet and our model on the same platform. The results demonstrate that our model has improved accuracy by 0.48% and 0.16% on CIFAR-10 and CIFAR-100, respectively. For FGVC-Aircraft and Stanford Cars datasets, we connect a MAMS-block after each residual block and raise the convolution kernel of $Conv5$ and $Conv6$ in Figure 4 to 256 and 512. The parameter quantities of the two networks reached 31.59M and 31.64M,

respectively. As can be seen from the test results, although we did not specifically study fine-grained images, we still achieved good results on two fine-grained datasets.

5. Conclusions

Attention mechanisms and multi-scale features are two important measures for dealing with computer vision tasks. AMS-block first uses channel attention to arrange the feature maps in descending order, then performs two residual convolutions on the feature maps. It subtly integrates the attention mechanism and multi-scale features into the convolution model for image classification. At a more detailed level of research, we propose DAMS-block and MAMS-block. Based on this, the novel convolution model we construct can not only focus on important features and ignore disturbing information, but also enable significant features to have multi-scale expression through feature sorting. The classification results of MAMS-CNN on multiple image sets, including standard image sets and fine-grained image sets, demonstrate its powerful classification performance. On the other hand, MAMS-block is easy to graft into the deep learning model for end-to-end training, which is one of its advantages. However, more intuitive applications of attention mechanisms and multi-scale features are image tasks other than image classification, such as object detection, saliency analysis. Next we will try to introduce MAMS-block into further image processing models to maximize its value.

Author Contributions: X.W. guided the theoretical and experimental design; F.D. and C.X. participated in the environmental configuration and data collection; Y.Y. completed experimental design and paper writing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 61872231, 61701297, 61703267) and the Scientific Research Fund of Hunan Provincial Education Department(Grant No. 15C1241).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Res-block	Residual block
ResNet	Superimposed by multiple Res-blocks
SE-block	Squeeze-and-Excitation block
SE-CNN	Superimposed by multiple SE-block
Res2-block	Hierarchical residual-like connections within residual block
Res2-CNN	Superimposed by multiple Res2-blocks
AMS-block	A multi-scale convolutional block based on single attention
AMS-CNN	Superimposed by multiple AMS-blocks
DAMS-block	A multi-scale convolutional block based on dual attention
DAMS-CNN	Superimposed by multiple DAMS-blocks
MAMS-block	A multi-scale convolutional block based on multiple attention
MAMS-CNN	Superimposed by multiple MAMS-blocks

Appendix A

Algorithm 1: AMS-block algorithm

Require: Input feature maps(x), number of channels(n)
Define AMS_block(x, n):
    Use the SE-block to get the channels importance dictionary "dict"
    Sort the "dict" and record the index: Index ← tf.nn.top_k(dict, n)
    According to the record of index, y is sorted by x: y ← tf.batch_gather(y, index)
    Divide the sorted channels into k groups, denoted as [y_1, y_2, ..., y_k]
    Perform a convolution operation on the first group: z ← Conv(y_1)
    For i in (2, k) do:
        Add the feature maps after convolution and the current group: $y \leftarrow \text{Add}(y_i, z)$
        Perform a convolution operation on this group: $y' \leftarrow \text{Conv}(y)$
        Group each group along the channels: z ← Contant(z, y')
        Assign y to z: $z \leftarrow y'$
    End for
    Add the x and z by using the residual learning idea: z ← Add(z, x)
    Use point-wise convolution to get the feature maps: Block ← Conv(z)
    Return block

References

1. Cao, Y.J.; Jia, L.L.; Chen, Y.X.; Lin, N.; Yang, C.; Zhang, B.; Liu, Z.; Li, X.X.; Dai, H.H. Recent Advances of Generative Adversarial Networks in Computer Vision. *IEEE Access* **2019**, *7*, 14985–15006. [CrossRef]
2. Choi, J.; Kwon, J.; Lee, K.M. Real-Time Visual Tracking by Deep Reinforced Decision Making. *Comput. Vis. Image Underst.* **2018**, *171*, 10–19. [CrossRef]
3. Shen, D.H.; Zhang, Y.Z.; Henao, R.; Su, Q.L.; Carin, L. Deconvolutional Latent-Variable Model for Text Sequence Matching. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5438–5445.
4. Liu, J.; Ren, H.L.; Wu, M.L.; Wang, J.; Kim, H.j. Multiple Relations Extraction Among Multiple Entities in Unstructured Text. *Soft Comput.* **2018**, *22*, 4295–4305. [CrossRef]
5. Kim, G.; Lee, H.; Kim, B.K.; Oh, S.H.; Lee, S.Y. Unpaired Speech Enhancement by Acoustic and Adversarial Supervision for Speech Recognition. *IEEE Signal Process. Lett.* **2019**, *26*, 159–163. [CrossRef]
6. Deena, S.; Hasan, M.; Doulaty, M.; Saz, O.; Hain, T. Recurrent Neural Network Language Model Adaptation for Multi-Genre Broadcast Speech Recognition and Alignment. *IEEE/ACM Trans. Audio Speech Lang.* **2019**, *27*, 572–582. [CrossRef]
7. Xie, J.J.; Li, A.Q.; Zhang, J.G.; Cheng, Z.A. An Integrated Wildlife Recognition Model Based on Multi-Branch Aggregation and Squeeze-And-Excitation Network. *Appl. Sci.* **2019**, *9*, 2749. [CrossRef]
8. Yang, Y.D.; Wang, X.F.; Zhao, Q.; Sui, T.T. Two-Level Attentions and Grouping Attention Convolutional Network for Fine-Grained Image Classification. *Appl. Sci.* **2019**, *9*, 1939. [CrossRef]
9. Li, Z.L.; Dong, M.H.; Wen, S.P.; Hu, X.; Zhou, P.; Zeng, Z.G. CLU-CNNs: Object Detection for Medical Images. *Neurocomputing* **2019**, *350*, 53–59. [CrossRef]
10. Jiang, Y.; Peng, T.T.; Tan, N. CP-SSD: Context Information Scene Perception Object Detection Based on SSD. *Appl. Sci.* **2019**, *9*, 2785. [CrossRef]
11. Yang, J.F.; Liang, J.; Shen, H.; Wang, K.; Rosin, P.L.; Yang, M.H. Dynamic Match Kernel with Deep Convolutional Features for Image Retrieval. *IEEE Trans. Image Process.* **2018**, *27*, 5288–5301. [CrossRef]
12. Yang, X.; Wang, N.N.; Song, B.; Gao, X.B. BoSR: A CNN-Based Aurora Image Retrieval Method. *Neural Netw.* **2019**, *116*, 188–197. [CrossRef] [PubMed]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet Classification with Deep Convolutional Neural Networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.

14. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
15. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
16. Itti, L.; Koch, C.; Niebur, E. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [CrossRef]
17. Itti, L.; Koch, C. Computational Modelling of Visual Attention. *Nat. Rev. Neurosci.* **2001**, *2*, 194–203. [CrossRef]
18. Meur, O.E.; Callet, P.L.; Barba, D.; Thoreau, D. A Coherent Computational Approach to Model Bottom-Up Visual Attention. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 802–817. [CrossRef]
19. Corbetta, M.; Shulman, G.L. Control of Goal-Directed and Stimulus-Driven Attention in the Brain. *Nat. Rev. Neurosci.* **2002**, *3*, 201–215. [CrossRef]
20. Baluch, F.; Itti, L. Mechanisms of Top-Down Attention. *Trends Neurosci.* **2011**, *34*, 210–224. [CrossRef]
21. Zhang, J.M.; Bargal, S.A.; Lin, Z.; Brandt, J.; Shen, X.H.; Sclaroff, S. Top-Down Neural Attention by Excitation Backprop. *Int. J. Comput. Vis.* **2018**, *126*, 1084–1102. [CrossRef]
22. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 7132–7141.
23. Yang, Y.D.; Wang, X.F.; Zhang, H.Z. Local Importance Representation Convolutional Neural Network for Fine-Grained Image Classification. *Symmetry* **2018**, *10*, 479. [CrossRef]
24. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 3–19.
25. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2Net: A New Multi-Scale Backbone Architecture. *arXiv* **2019**, arXiv:1904.01169.
26. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
27. Lin, M.; Chen, Q.; Yan, S.C. Network In Network. *arXiv* **2014**, arXiv:1312.4400.
28. Szegedy, C.; Liu, W.; Jia, Y.Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1–9.
29. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. PRethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
30. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
31. Chollet, F. Xception: Deep Learning With Depthwise Separable Convolutions. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
32. Howard, A.G.; Zhu, M.L.; Chen, B.; Kalenichenko, D.; Wang, W.J.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
33. Zhang, X.Y.; Zhou, X.Y.; Lin, M.X.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 6848–6856.
34. Zhang, T.; Qi, G.J.; Xiao, B.; Wang, J.D. Interleaved Group Convolutions for Deep Neural Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4373–4382.
35. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the International Conference on Learning Representations, Caribe Hilton, San Juan, Puerto Rico, 2–4 October 2016; pp. 1–13.
36. Dai, J.F.; Qi, H.Z.; Xiong, Y.W.; Li, Y.; Zhang, G.D.; Hu, H.; Wei, Y.C. Deformable Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.

37. Li, X.; Wang, W.H.; Hu, X.L.; Yang, J. Selective Kernel Networks. *arXiv* **2019**, arXiv:1903.06586.
38. Rupesh, K.S.; Klaus, G.; Jürgen, S. Highway Networkss. *arXiv* **2015**, arXiv:1505.00387v2.
39. Sergey, Z.; Nikos, K. Wide Residual Networks. *arXiv* **2017**, arXiv:1605.07146v4.
40. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
41. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–27.
42. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
43. Liu, S.; Qi, L.; Qin, H.F.; Shi, J.P.; Jia, J.Y. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 8759–8768.
44. Lin, D.; Shen, D.G.; Shen, S.T.; Ji, Y.F.; Lischinski, D.N.; Cohen-Or, D.; Huang, H. ZigZagNet: Fusing Top-Down and Bottom-Up Context for Object Segmentation. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7490–7499.
45. Li, W.B.; Wang, Z.C.; Yin, B.Y.; Peng, Q.X.; Du, Y.M.; Xiao, T.Z.; Yu, G.; Lu, H.T.; Wei, Y.C.; Sun, J. Rethinking on Multi-Stage Networks for Human Pose Estimation. *arXiv* **2019**, arXiv:1901.00148.
46. Zhao, Q.J.; Sheng, T.; Wang, Y.T.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H.B. M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 1–8.
47. Xiao, T.T.; Liu, Y.C.; Zhou, B.L.; Jiang, Y.N.; Sun, J. Unified Perceptual Parsing for Scene Understanding. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September **2018**; pp. 1–17.
48. Yang, L.; Song, Q.; Wang, Z.H.; Jiang, M. Parsing R-CNN for Instance-Level Human Analysis. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 364–373.
49. Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv* **2013**, arXiv:1306.5151.
50. Krause, J.; Stark, M.; Jia, D.; Li, F.F. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 3–6 December 2013; pp. 554–561.
51. Gosselin, P.H.; Murray, N.; Jégou, H.; Perronnin, F. Revisiting the Fisher vector for fine-grained classification. *Pattern Recogn. Lett.* **2014**, *49*, 92–98. [CrossRef]
52. Zhao, B.; Wu, X.; Feng, J.S.; Peng, Q.; Yan, S.C. Diversified Visual Attention Networks for Fine-Grained Object Classification. *IEEE Trans. Multimed.* **2017**, *19*, 1245–1256. [CrossRef]
53. Gao, Y.; Beijbom, O.; Zhang, N.; Darrell, T. Compact Bilinear Pooling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 317–326.
54. Kong, S.; Fowlkes, C. Low-Rank Bilinear Pooling for Fine-Grained Classification. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 365–374.