Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

# HAM: Hybrid attention module in deep convolutional neural networks for image classification

Guoqiang Li[a,b,*], Qi Fang[a,b], Linlin Zha[a,b], Xin Gao[a,b], Nenggan Zheng[c]

[a] *School of Electrical Engineering, Yanshan University, No.438 West Hebei Avenue, Qinhuangdao 066000, China*
[b] *Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066000, China*
[c] *Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, China*

ARTICLE INFO

*Article history:*
Received 17 August 2021
Revised 13 March 2022
Accepted 7 May 2022
Available online 10 May 2022

*Keywords:*
Hybrid attention module
Channel attention map
Spatial feature descriptor
HAM-integrated networks

ABSTRACT

Recently, many researches have demonstrated that the attention mechanism has great potential in improving the performance of deep convolutional neural networks (CNNs). However, the existing methods either ignore the importance of using channel attention and spatial attention mechanisms simultaneously or bring much additional model complexity. In order to achieve a balance between performance and model complexity, we propose the Hybrid Attention Module (HAM), a really lightweight yet efficient attention module. Given an intermediate feature map as the input feature, HAM firstly produces one channel attention map and one channel refined feature through the channel submodule, and then based on the channel attention map, the spatial submodule divides the channel refined feature into two groups along the channel axis to generate a pair of spatial attention descriptors. By applying saptial attention descriptors, the spatial submodule generates the final refined feature which can adaptively emphasize the important regions. Besides, HAM is a simple and general module, it can be embedded into various mainstream deep CNN architectures seamlessly and can be trained with base CNNs in the end-to-end way. We evaluate HAM through abundant of experiments on CIFAR-10, CIFAR-100 and STL-10 datasets. The experimental results show that HAM-integrated networks achieve accuracy improvements and further reduce the negative impact of less training data on deeper networks performance than its counterparts, which proves the effectiveness of HAM.

## 1. Introduction

Convolutional neural networks (CNNs) have been widely used in computer vision tasks due to their powerful representation ability [1,2], which were inspired by biological natural vision cognitive mechanisms [3]. In recent years, more and more researchers have focused on deeper and wider CNNs to meet the requirements of visual tasks [4,5]. One of the most important reasons is that deep CNN architectures can produce advanced and abstract features as the networks go deeper [6]. Another reason which makes deep networks become possible is the rapid development of GPU [7].

Starting with AlexNet [8], a deep convolutional network with eight layers won the championship of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, which made deep convolution architectures come into view. Since then, deep convolution architectures have been continuously investigated to further improve models' performance. VGGNet [9], which reached a depth of 19, shows that stacking identical blocks can get fair results. Network in Network [10] and GoogLeNet [11] have made great progress in layer design, which obtained a better performance with reducing the calculation and parameters of the model. However, many researchers found that when the network reaches a certain depth, it would degenerate. Things came to a turning point when *He* et al. created out Deep Residual Network [12]. This network uses identity mapping to overcome vanishing gradients and networks degeneration during training deep networks, which breaks through the layer number to a thousand. Nevertheless, many studies pointed out that networks performance does not grow so much as they go deeper, which resulted in wasting much more computing resource in training process. For this reason, some researchers came up with a new idea, making networks pay more attention to important features while suppressing unnecessary ones, which is named attention mechanism [13].

Attention mechanism significantly improves the performance of networks. It not only tells us where to focus, but also increases

[*] Corresponding author.
*E-mail addresses:* lig_ysu@163.com (G. Li), fangqi20201218@163.com (Q. Fang), amazinglyn@163.com (L. Zha), gao1259944745@163.com (X. Gao), zng@cs.zju.edu.cn (N. Zheng).

https://doi.org/10.1016/j.patcog.2022.108785

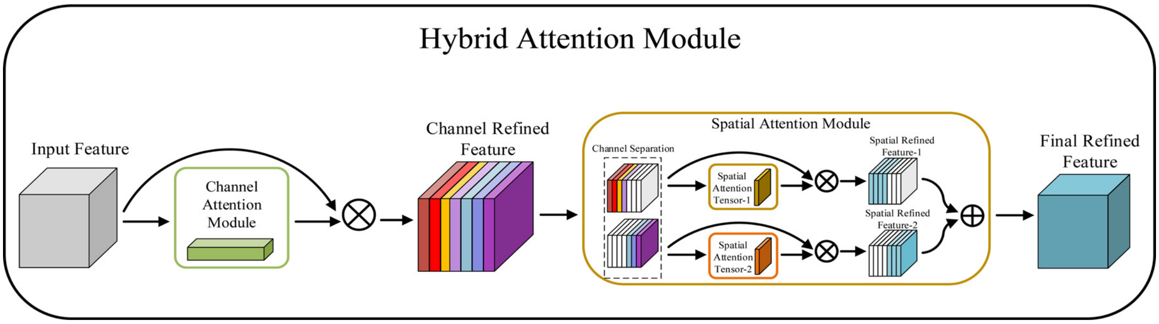

**Fig. 1.** The structure of HAM. It consists of two submodules in sequential manner, *channel* and *spatial* modules. Channel modules could produce a 1*D* attention map while spatial modules produce a pair of 2*D* attention maps. The intermediate features could get refined in channel and spatial-wise through HAM which can be embedded into any state-of-the-art deep CNNs.

representation power of interests [14]. Squeeze-and-Excitation Network (SENet) [15], one of the representative networks based on channel attention, which brings obvious performance improvement for deep CNNs. Based on the operation, squeeze-and-excitation in SENet, some researchers have improved this block by seeking for better channel-wise dependencies [16–18]. Spatial attention is another important attention mechanism, aiming at finding the significance parts in final targets, which can refine the networks to get higher accuracy. CCNet [19] is a state-of-the-art and representative spatial attention network that applies criss-cross attention to capture full-image contextual information, resulting in improving network performance. Furthermore, Convolutional Block Attention Module (CBAM) [20] has combined channel attention with spatial attention in their tasks to get impressive results. Although these methods can achieve excellent performance, they always bring extra complexity and computation of models. In order to get rid of higher model complexity, this paper proposes a Hybrid Attention Module (HAM), a lightweight module, which can be embedded into existing state-of-the-art CNNs. Since both channel attention and spatial attention are all important to final targets, we adopt our module by sequentially applying channel-wise and spatial-wise attention as shown in Fig. 1. As a result, our module helps networks efficiently learn which features to emphasize or suppress.

The main contributions of this paper can be generalized as follows:

(1) we design a lightweight yet efficient attention module, Hybrid Attention Module, which can be widely used in deep CNNs to improve networks performance.

(2) we conduct ablation experiments to validate the effectiveness of our attention module.

(3) we embed our attention module into mainstream convolution architectures and achieve performance superior to the state-of-the-art.

(4) we further carry out experiments on the STL-10 dataset with ResNet [12] of different depths, demonstrating that HAM can better reduce the negative impact of less training data on the performance as networks go deeper when compared with other attention modules.

## 2. Related works

**Network engineering.** In the past few years, many researchers were keep on “Network engineering” that deeper networks were designed to ensure significant performance improvement in various vision tasks [21]. However, deep networks were very hard to train due to the difficulty of gradient back propagation. Residual Networks [12,22], which provides the design philosophy of identity skip-connection, have been considered to be very effective in training deep architectures. From that time on, a large number of deep networks have been proposed based on this idea. Inception-ResNet [23] which introduces residual architecture into inception blocks obtains state of art accuracy. Wide Residual Network (WRN) [24] is carried out with more filters and fewer layers which outperforms in representation ability. PyramidNet [25] is a generalization of WRN with strict rules that channels in networks should gradually increase. ResNeXt [26] suggests that increasing the cardinality could win better image classification accuracy than increasing channels. In the last several years, *Huang* et al. proposed a new network named DenseNet [27]. It concatenates the input features with output features iteratively by using densely connected residual architectures, which proves that features from previous layers can be reused in deep CNN architectures through skip-connection. While the methods of current network engineering primarily concentrate on three factors: depth, width and cardinality, we focus on another aspect, attention mechanisms.

**Attention mechanisms.** Evidence from human visual system shows that attention mechanisms play an important role in capturing significant information. Inspired by this phenomenon, many researchers attempted to incorporate attention mechanisms to enhance the performance of deep CNNs in image classification tasks [28,29]. *Wang* et al. proposed Residual Attention Network which is constructed by stacking encoder-decoder style attention modules [30]. The network not only performs well, but also is robust to noisy inputs. *Hu* et al. showed Squeeze-and-Excitation Network (SENet) to us, which was designed to explore inter-channel relationship [15]. Based on this work, *Wang* et al. proposed Efficient Channel Attention (ECA) [31] to achieve the relationship between different channels by one-dimensional convolution. *Huang* et al. proposed CCNet [19], which is a representative spatial attention network based on applying criss-cross attention to capture full-image contextual information.

Closer to our work, *Woo* et al. introduced Convolutional Block Attention Module (CBAM) [20] to obtain channel attention and spatial attention at the same time. In their attention module, they apply a multi-layer perceptron with one hidden layer to compute channel refined features. Then the spatial submodule produces a 2*D* spatial attention map based on channel refined features and multiplies the spatial attention map to channel refined features to get final refined features. However, this attention module is accompanied by redundant computation and model complexity while achieving higher accuracy. Furthermore, we argue that the channel attention map represents that different importance different channel is, so that the spatial attention submodule should divide the channel refined features into groups along the channel axis. In our module(HAM), we redesign both channel attention and spatial attention as shown in Fig. 1. For the channel anttention submodule, we develop an adaptive mechanism between global average-pooled features and global max-pooled features due to their dif-

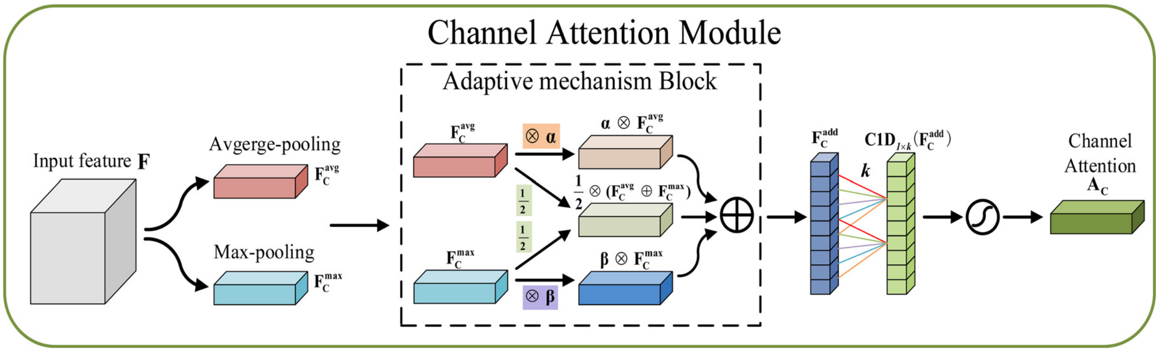

**Fig. 2.** Diagram of channel attention submodule of HAM. We design an adaptive mechanism to obtain enriched features as average-pooled and max-pooled features play different roles in channel attention inference. Then, the enriched features flow into the fast one-dimensional convolution and are further activated by the sigmoid function to generate the final channel attention.

ferent roles in different stages of images feature extraction. For the spatial submodule, we exploit channel separating operation to divide the channel refined feature into two groups along the channel axis, which is based on channel attention maps. Moreover, our module is general and can be embedded into any mainstream deep CNN achitectures. The experimental results show that our module can achieve state-of-the-art performance in classification tasks on CIFAR-10, CIFAR-100 and STL-10 datasets, which demonstrates the effectiveness of HAM. Especially, on STL-10 dataset which has far less training data than CIFAR datasets, HAM can weaken the negative impact of less training data on the performance when networks go deeper.

## 3. Methodology

Suppose $\mathbf{F} \in \mathbb{R}^{H\times W\times C}$ is an intermediate feature map as input tensor, HAM first derives a $1D$ channel attention tensor $\mathbf{A_C} \in \mathbb{R}^{1\times 1\times C}$ and multiplies it to the original input tensor to get the channel refined feature. Then, based on the channel attention map, the spatial attention submodule separates the channel refined feature into two groups along the channel axis. These two groups of features produce their own $2D$ spatial attention tensors: $\mathbf{A_{S,1}} \in \mathbb{R}^{H\times W\times 1}$ and $\mathbf{A_{S,2}} \in \mathbb{R}^{H\times W\times 1}$ as illustrated in Fig. 1. By multiplying the pair of spatial attention maps to their own group features respectively, the spatial attention submodule would generate a pair of spatial refined features. Ultimately, the final refined feature is obtained by adding the pair of spatial refined features up. This process can be summarized by the following equations:

$$\mathbf{F}' = \mathbf{A_C}(\mathbf{F}) \otimes \mathbf{F} \tag{1}$$

$$\mathbf{F}'_1 \oplus \mathbf{F}'_2 = \mathbf{F}' \tag{2}$$

$$\begin{aligned} \mathbf{F}''_1 &= \mathbf{A_{S,1}}(\mathbf{F}'_1) \otimes \mathbf{F}'_1 \\ \mathbf{F}''_2 &= \mathbf{A_{S,2}}(\mathbf{F}'_2) \otimes \mathbf{F}'_2 \end{aligned} \tag{3}$$

$$\mathbf{F}'' = \mathbf{F}''_1 \oplus \mathbf{F}''_2 \tag{4}$$

where $\otimes$ means element-wise multiplication and $\oplus$ means element-wise summation. In the process of multiplication, $\mathbf{F}'$ is the result of channel attention values being broadcasted along the spatial dimension and $\mathbf{F}''_i$ is the spatial refined feature which is also the result of spatial attention values being broadcasted along channel dimension. During summation, the channel refined feature $\mathbf{F}'$ is separated into two parts along the channel axis, $\mathbf{F}'_1$ and $\mathbf{F}'_2$. $\mathbf{F}''$ is the final output which is also the sum of $\mathbf{F}''_1$ and $\mathbf{F}''_2$. Fig. 2 and Fig. 3 depicts the calculation process of channel and spatial attention maps, respectively. The details of each attention module are elaborated as follows.

**Channel attention module.** Due to the importance of relationship between channels, we design a channel attention module. Channel attention concentrates on 'what' is more significant in input images for the reason that channels of feature maps are considered as feature detectors. In order to aggregate spatial dimension information, *Zhou* et al. suggested that using average-pooling can be convenient and efficient [32]. Soon, SENet [15] used it in their attention module. Based on this philosophy, *Woo* et al. demonstrated that max-pooling can produce another important clue in achieving channel attention [20]. As a result, they employing both average-pooled and max-pooled features simultaneously. Beyond their works, we argue that average-pooled and max-pooled features play different roles in different stages of images feature extraction. Thus, we exploit an adaptive mechanism in average-pooled and max-pooled features as they should not be weighted as the same. Furthermore, we use one-dimensional convolution to avoid dimensionality reduction of channels and capture cross-channel interaction. We verify that our module outperforms aforementioned methods, which shows the effectiveness of our design scheme. The operation is described in detail below.

As shown in Fig. 2, first, we use both average-pooling and max-pooling operations to aggregate spatial dimension information. The operations generate two different feature tensors, $\mathbf{F_C^{avg}}$ and $\mathbf{F_C^{max}}$, which stand for average-pooled and max-pooled features respectively. Then, both of two tensors are sent to the adaptive mechanism block to obtain enriched features $\mathbf{F_C^{add}} \in \mathbb{R}^{1\times 1\times C}$. The adaptive mechanism block has two trainable parameters $\alpha$ and $\beta$, which can be trained by Stochastic Gradient Descent (SGD). Eq. (5) sums up the inner process of this adaptive mechanism block.

$$\mathbf{F_C^{add}} = \frac{1}{2} \otimes \left(\mathbf{F_C^{avg}} \oplus \mathbf{F_C^{max}}\right) \oplus \alpha \otimes \mathbf{F_C^{avg}} \oplus \beta \otimes \mathbf{F_C^{max}} \tag{5}$$

where $\alpha$ and $\beta$ are both floating numbers greater than zero and less than one. These two trainable parameters not only produce the adaptive mechanism between average-pooled and max-pooled features but also enrich the features in the process of images feature extraction.

In SENet [15] and CBAM [20], they use a multi-layer perceptron (MLP) with one hidden layer to compute the channel attention. Consequently, their channel attention module brings additional model complexity and is affected by channel dimensionality reduction simultaneously. To address the limitations mentioned above, we make use of a fast one-dimensional convolution. For capturing interaction between channels, we set the kernel size of one-dimensional convolution to $k$. The parameter $k$ represents interaction between $k$ neighbors. To determine the value of $k$, we use

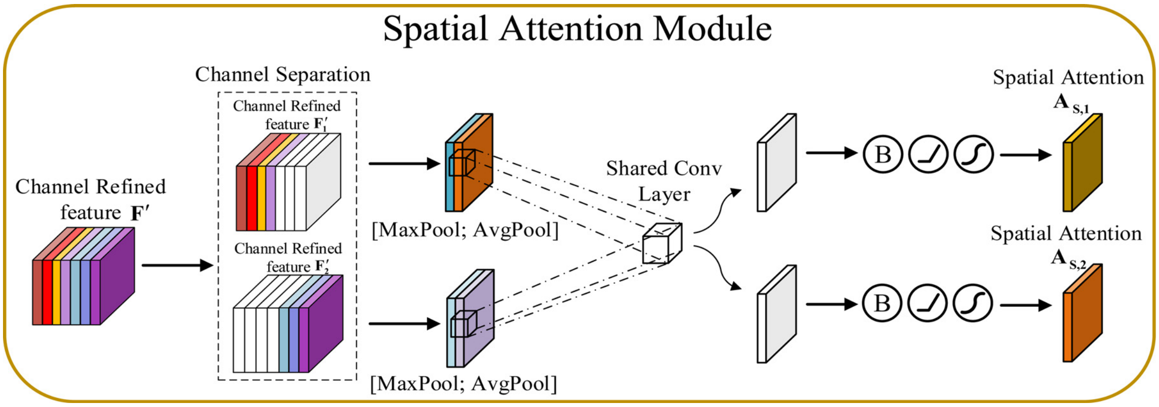
Spatial Attention Module
Channel Separation
Channel Refined feature $\mathbf{F}'_1$
Channel Refined feature $\mathbf{F}'$
Channel Refined feature $\mathbf{F}'_2$
[MaxPool; AvgPool]
Shared Conv Layer
[MaxPool; AvgPool]
Spatial Attention $\mathbf{A}_{S,1}$
B
Spatial Attention $\mathbf{A}_{S,2}$
B

**Fig. 3.** Diagram of spatial attention submodule of HAM. We exploit channel separation technique in the spatial attention submodule, which divides the channel refined feature into two groups based on the channel attention map. Then, the spatial attention submodule performs average-pooling and max-pooling operations on both sets of features along the channel axis and forward the ouputs to a shared convolution layer.

empirical Eq. (6) in ECA [31].

$$k = \phi(\mathbf{C}) = \left| \frac{\log_2(\mathbf{C})}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{6}$$

where $|t|_{odd}$ denotes the nearest odd number of $t$ and $\mathbf{C}$ is the number of channels. $\gamma$ and $b$ are both hyper-parameters and we generally set them to 2 and 1 respectively in all experiments of this paper. Through this mapping $\phi$, kernel size $k$ can be adaptively determined by the number of channels $\mathbf{C}$.

After the feature map $\mathbf{F}_{\mathbf{C}}^{\mathbf{add}} \in \mathbb{R}^{1\times1\times C}$ flows through the one-dimensional convolution, we employ sigmoid function to activate the output feature tensors. In a few words, the process of channel attention computing can be generalized as Eq. (7).

$$\begin{aligned}\mathbf{A_C}(\mathbf{F}) &= \sigma\Big(\mathbf{C1D}_{1\times k}\Big(\frac{1}{2} \otimes (AvgPool(\mathbf{F}) \oplus MaxPool(\mathbf{F})) \oplus \alpha \otimes AvgPool(\mathbf{F}) \oplus \beta \otimes MaxPool(\mathbf{F})\Big)\Big)\\ &= \sigma\Big(\mathbf{C1D}_{1\times k}\Big(\frac{1}{2} \otimes (\mathbf{F}_{\mathbf{C}}^{\mathbf{avg}} \oplus \mathbf{F}_{\mathbf{C}}^{\mathbf{max}}) \oplus \alpha \otimes \mathbf{F}_{\mathbf{C}}^{\mathbf{avg}} \oplus \beta \otimes \mathbf{F}_{\mathbf{C}}^{\mathbf{max}}\Big)\Big)\\ &= \sigma\big(\mathbf{C1D}_{1\times k}(\mathbf{F}_{\mathbf{C}}^{\mathbf{add}})\big)\end{aligned} \tag{7}$$

where $\sigma$ represents the sigmoid function and $\mathbf{C1D}_{1\times k}$ stands for one-dimensional convolution with kernel size of $k$.

**Spatial attention module.** Channel attention concentrates on 'what' is more important while spatial attention focuses on 'where' is more important. In the channel attention submodule, a channel attention map is produced, which shows different channels have different importance. This clearly tells us that every channel has a different importance in the channel refined feature. To be more exact, the channel attention map assigns larger weights to important channels of the feature map while assigning smaller weights to sub-important channels. As a result, we should partition the channel refined feature along the channel axis based on the channel attention map, which is named channel separation technique by us. Before further describing spatial attention, we introduce the channel separation technique.

The channel separation technique will divide the channel refined feature into two groups along the channel axis based on the channel attention map. Thus, it has a hyperparameter, separation rate $\lambda$, which is the boundary between important and sub-important groups of channels. First, we multiply $\lambda$ to the channel dimension of the channel refined feature to obtain the channel dimension of the important feature. In addition, it is well known that the channel dimension usually is an even number. Therefore, we take the nearest even number of the channel dimension of the important feature, which is marked as $\mathbf{C}_{im}$. The computing process of $\mathbf{C}_{im}$ can be written as Eq. (8).

$$\mathbf{C}_{im} = |\mathbf{C}_{C-R} \cdot \lambda|_{even} \tag{8}$$

where $\mathbf{C}_{C-R}$ is the channel dimension of the channel refined feature and $|t|_{even}$ denotes the nearest even number of $t$.

$\mathbf{C}_{im}$ can make sure the channel dimension of important and sub-important features are both even numbers. What's more, $\mathbf{C}_{im}$ can determine top $n$ maximum values in the channel attention map, which correspond to top $n$ important channels in the channel refined feature. Note that $\mathbf{C}_{im}$ and $n$ are numerically equal but their physical meanings are different. After determining the value of $n$, we can obtain the $n$th largest value in the channel attention map. At this point, we can think of the values in the channel attention map as two parts. One part is that values are greater than or equal to the $n$th largest value and we name this part with important part. The other part is that values are less than the $n$th largest value and we name it with sub-important part. Thus, we could define two masks and both of them have the same shape as the channel attention map. One mask is that we let the values in important part take one while the values in sub-important part take zero. The other mask is the opposite. We name these masks with important mask and sub-important mask respectively. Then, by multiplying both of masks to the channel refined feature, we naturally divide the channel refined feature into important group of channels $\mathbf{F}'_1$ and sub-important group of channels $\mathbf{F}'_2$. As shown in Fig. 3, the channel refined feature $\mathbf{F}'_1$ and $\mathbf{F}'_2$ both have channels with zreo values which are marked by white cuboids. Obviously, the channel separation technique doesn't change channels order of the channel refined feature and satisfy the Eq. (2). Algorithm 1 gives pseudo-code of the channel separation technique.

Similar to the designed channel attention submodule, we perform average-pooling and max-pooling operations on both $\mathbf{F}'_1$ and $\mathbf{F}'_2$ in the direction of the channel axis and concatenate the outputs to generate two feature descriptors. It is obvious that this pair of feature descriptors corresponds to $\mathbf{F}'_1$ and $\mathbf{F}'_2$ respectively. On the pair of concatenated feature descriptors, we utilize a shared convolution layer to generate a pair of spatial attention maps: $\mathbf{A_{S,1}} \in \mathbb{R}^{H\times W\times 1}$ and $\mathbf{A_{S,2}} \in \mathbb{R}^{H\times W\times 1}$. The pair of spatial attention maps tells us where to emphasize or suppress. The inference process of spatial attention is described in detail below.

As shown in Figure 3, we first perform channel separation on the channel refined feature $\mathbf{F}'$ to divide it into the important group of channels $\mathbf{F}'_1$ and the sub-important group of channels $\mathbf{F}'_2$. Then, we use average-pooling and max-pooling operations to aggregate channel dimension information on both $\mathbf{F}'_1$ and $\mathbf{F}'_2$, generating two pairs of $2D$ maps. One pair is $\mathbf{F}_{\mathbf{S,1}}^{\mathbf{avg}} \in \mathbb{R}^{H\times W\times 1}$ and $\mathbf{F}_{\mathbf{S,1}}^{\mathbf{max}} \in \mathbb{R}^{H\times W\times 1}$, the other pair is $\mathbf{F}_{\mathbf{S,2}}^{\mathbf{avg}} \in \mathbb{R}^{H\times W\times 1}$ and $\mathbf{F}_{\mathbf{S,2}}^{\mathbf{max}} \in \mathbb{R}^{H\times W\times 1}$. Note that when we perform average-pooling operation on $\mathbf{F}'_1$, the value we

**Algorithm 1** Channel separation technique.

```
Input: $x$                                                        ▷ the channel refined feature
Require: $\mathbf{M}$                                             ▷ the channel attention map
Require: $\lambda$                                                ▷ separation rate
  $\mathbf{C} \leftarrow$ GETCHANNELDIMENSION(x)
  $h = \mathbf{C} \cdot \lambda$
  $\mathbf{C}_{im} \leftarrow$ THENEARSTEVEN($h$)                  ▷ the channel dimension of important features
  $\mathbf{C}_{subim} = \mathbf{C} - \mathbf{C}_{im}$              ▷ the channel dimension of sub-important features
  important_mask $\leftarrow$ TOP($\mathbf{M}, \mathbf{C}_{im}$)    ▷ top $\mathbf{C}_{im}$ important channels in the channel attention map
  nth_important $\leftarrow$ MIN(important_mask)                 ▷ the minimum value of *important_mask*
  subimportant_mask $\leftarrow$ LESS($\mathbf{M}$, nth_important) ▷ sub-important channels in the attention map
  important_channels = $\mathbf{M}$
  subimportant_channels = $\mathbf{M}$                          ▷ create tensors equal to the channel attention map

  for $i$ in range($\mathbf{C}$) do
    if $\mathbf{M}[i] \in$ important_mask then
      important_channels[$i$] = 1
    else
      important_channels[$i$] = 0
    end if
  end for
  return important_channels                                     ▷ return the mask of important channels

  for $i$ in range($\mathbf{C}$) do
    if $\mathbf{M}[i] \in$ subimportant_mask then
      important_channels[$i$] = 1
    else
      subimportant_channels[$i$] = 0
    end if
  end for
  return subimportant_channels                                  ▷ return the mask of sub-important channels

  important_features $\leftarrow$ MULTIPLY($x$, important_channels)       ▷ the important features
  subimportant_features $\leftarrow$ MULTIPLY($x$, subimportant_channels) ▷ the sub-important features
Output: important_features
Output: important_features
```

divide by is $\mathbf{C}_{im}$. Naturally, when we perform average-pooling operation on $\mathbf{F}'_2$, the value is $\mathbf{C}_{C-R} - \mathbf{C}_{im}$. After pooling operations, we concatenate the outputs to produce a pair of feature descriptors. Next, the pair of concatenated feature descriptors is convolved by a shared convolution layer with kernel size of $7 \times 7$ to generate a pair of $2D$ attention maps. Finally, $2D$ attention maps should go through normalization and activation operations to generate our spatial attention maps: $\mathbf{A_{S,1}} \in \mathbb{R}^{H \times W \times 1}$ and $\mathbf{A_{S,2}} \in \mathbb{R}^{H \times W \times 1}$. In short words, the process of the spatial attention computation can be summarized as Eq. (9) and Eq. (10).

$$\begin{aligned} \mathbf{A_{S,1}}(\mathbf{F}') &= \Phi\left(\mathbf{C2D}_{7\times7}\left(\left[AvgPool(\mathbf{F}'_1); MaxPool(\mathbf{F}'_1)\right]\right)\right) \\ &= \Phi\left(\mathbf{C2D}_{7\times7}\left(\left[\mathbf{F^{avg}_{S,1}}; \mathbf{F^{max}_{S,1}}\right]\right)\right) \end{aligned} \tag{9}$$

$$\begin{aligned} \mathbf{A_{S,2}}(\mathbf{F}') &= \Phi\left(\mathbf{C2D}_{7\times7}\left(\left[AvgPool(\mathbf{F}'_2); MaxPool(\mathbf{F}'_2)\right]\right)\right) \\ &= \Phi\left(\mathbf{C2D}_{7\times7}\left(\left[\mathbf{F^{avg}_{S,2}}; \mathbf{F^{max}_{S,2}}\right]\right)\right) \end{aligned} \tag{10}$$

where $\Phi$ denotes a series of nonlinear operations, in order of the batch normalization and the ReLU function and the sigmoid function. The ReLU function can eliminate negative elements of spatial attention maps to focus only on the features that have a positive impact on the final classification result. $\mathbf{C2D}_{7\times7}$ represents a shared convolution layer with kernel size of $7 \times 7$.

**Arrangement of attention module.** The channel and spatial attention compute complementary attention information, concentrating on 'what' and 'where' need to emphasize respectively. Our attention module (HAM) can be embedded into any deep CNN architectures to significantly improve networks performance. Fig. 4 shows a diagram of HAM embedded into a ResBlock in ResNet [12] as an example.

## 4. Experiments

In order to evaluate HAM, we conduct experiments on standard datasets: CIFAR-10, CIFAR-100 and STL-10. Firstly, we do abundant of ablation experiments which show the validity of the design scheme of HAM. Secondly, we embed HAM into different network architectures and achieve better performance compared to other attention modules mentioned above, which demonstrates the effectiveness of HAM. Thirdly, we further conduct experiments on the STL-10 dataset with ResNet [12] of different depths as backbone networks, showing that HAM can reduce the negative impact in the case of less training data as networks go deeper. For better comparisons, we reproduce all the experimental networks in TensorFlow framework [33] and apply the reproduced results in the whole experiments.

### *4.1. Ablation studies*

In these ablation studies, we apply CIFAR datasets and choose ResNet-56 [12] as the backbone model of attention modules. CIFAR classification datasets are widely used in evaluating the representation of networks. CIFAR datasets have two kinds, CIFAR-10 and CIFAR-100. Both of them consist of 60,000 images with the size of $32 \times 32$. Among these images, 50,000 images are applied for training and 10,000 images are applied for test. CIFAR-10 has 10 classes

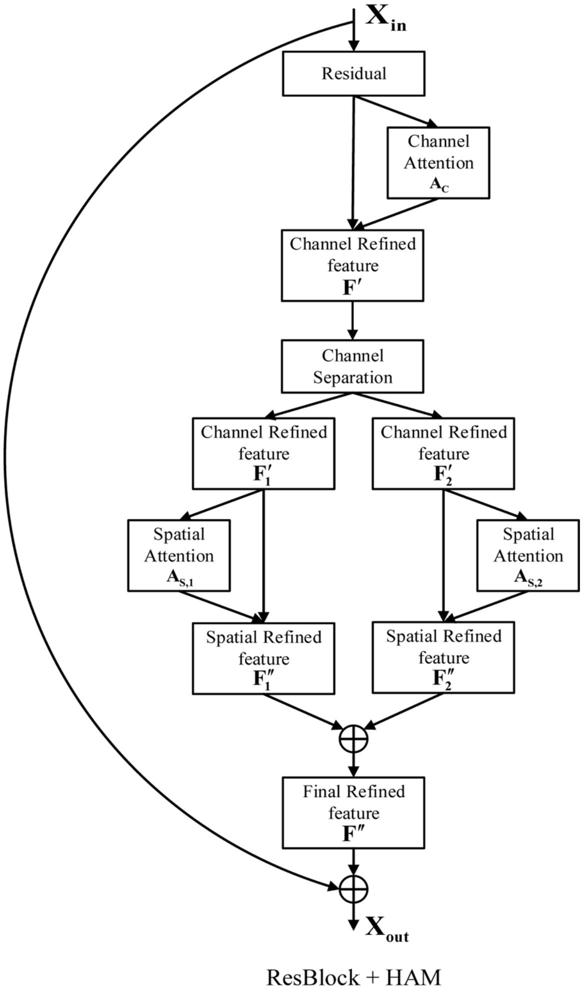
$\mathbf{X}_{in}$
Residual
Channel Attention $\mathbf{A}_C$
Channel Refined feature $\mathbf{F}'$
Channel Separation
Channel Refined feature $\mathbf{F}'_1$
Channel Refined feature $\mathbf{F}'_2$
Spatial Attention $\mathbf{A}_{S,1}$
Spatial Attention $\mathbf{A}_{S,2}$
Spatial Refined feature $\mathbf{F}''_1$
Spatial Refined feature $\mathbf{F}''_2$
Final Refined feature $\mathbf{F}''$
$\mathbf{X}_{out}$
ResBlock + HAM

**Fig. 4.** Diagram of HAM integrated into a ResBlock in ResNet [12]. This figure clearly indicates the exact location of our HAM when it is embedded into a ResBlock. Our HAM can be embedded into each block.

while CIFAR-100 has 100 classes. For training, we use the standard data augmentation: each side pads 4 pixels and the padded images are randomly sampled with a 32 × 32 crop or random horizontal flip. For test, we only use original 32 × 32 images. All models are trained with a mini-batch size of 128 on one GPU. We use a weight decay of 0.0005 and a momentum of 0.9. We set the initial learning rate to 0.1 and divide it by 5 at 60th, 120th and 160th epochs. We train the models for 200 epochs and report classification errors on test data.

Our ablation experiments can be divided into three parts. We first validate the effectiveness of channel attention module and then spatial attention module. Finally, we explore a better value for separation rate $\lambda$ to promote our HAM to achieve the best performance. The experimental details are introduced as follows.

**Channel attention.** In this part, we show that our attention module has better attention inference power through ablation experiments. We compare our channel attention module with other 3 channel attention modules: SE module [15], ECA module [31] and the channel attention submodule in CBAM [20]. Besides, we add two other variant modules for comparison: the channel attention submodule in CBAM with the proposed adaptive mechanism and the channel attention submodule in our HAM without the adaptive mechanism. Note that there is a reduction ratio in both SE module and channel attention module in CBAM. To perform better comparisons, we set them all to 16. Moreover, $\alpha$ and $\beta$, the two trainable parameters of our adaptive mechanism, are both initialized to 0.5. For the convenience of labeling, channel attention in CBAM is named CBAM (channel), channel attention in CBAM with the adaptive mechanism is named CBAM (channel, adaptive), channel attention in our HAM without the adaptive mechanism is named HAM (channel, not adaptive) and the original channel attention in our HAM is named HAM (channel).

The experimental results with various channel attention are shown in Table 1. Comparing the accuracy of CBAM [20] (channel) and CBAM (channel, adaptive), HAM (channel, not adaptive) and HAM (channel), we can clearly observe that our proposed adaptive mechanism plays an important role in channel attention computing. In CBAM (channel) and HAM (channel, not adaptive), they only consider using both average-pooled and max-pooled features, missing the inequality of the importance of these two features. We hold that the adaptive mechanism between these two features can enrich the feature map and allow them to have a complementary relationship. Therefore, we suggest to use the designed adaptive mechanism between average-pooled and max-pooled features. Beyond this work, we apply a fast one-dimensional convolution with kernel size of $k$ instead of two fully connected layers, which can bring less model complexity and avoid channel dimensionality reduction. As shown in Table 1, compared to CBAM (channel, adaptive), HAM (channel) achieves a 4.3% improvement in error rate with bring less parameters (about 1.1%) and less memory (about 1.2%) on CIFAR-10 dataset. Throughout the whole Table 1, our attention module, HAM (channel), achieves the highest accuracy without bring much extra parameters and computations. Thus, we recommend to use both an adaptive mechanism and a fast one-dimensional convolution with kernel size of $k$ to these features. The results objectively show that our channel attention design scheme is effective in channel attention inference, so that we use our attention module in the following experiments.

**Spatial attention.** After the channel attention module, we have to exploit an effective approach to figure out the spatial attention. Since the channel attention map indicates to us that different channels have different levels of importantce, we first use our proposed channel separation technique to divide the channel-wise refined feature into two groups of channels. Notice that we set

**Table 1**
Comparison of various channel attention modules. This table shows that applying our designed HAM(channel) outperforms other channel attention submodules. Besides, the channel attention module in CBAM [20] with our proposed adaptive mechanism outperforms the original channel attention module in CBAM. Moreover, the channel attention module in HAM without the adaptive mechanism doesn't perform as well as the original one.

| Description | Parameters | GFLOPs | Memory(M) | CIFAR-10 Error(%) | CIFAR-100 Error(%) |
|---|---|---|---|---|---|
| ResNet56 [12] (baseline) | 0.85M | 0.25 | 3.24 | 7.24 | 31.85 |
| ResNet56 + SE [15] | 0.86M | 0.25 | 3.28 | 6.84 | 30.94 |
| ResNet56 + ECA [31] | 0.85M | 0.25 | 3.24 | 6.55 | 30.45 |
| ResNet56 + CBAM [20] (channel) | 0.86M | 0.25 | 3.28 | 6.77 | 30.87 |
| ResNet56 + CBAM (channel, adaptive) (ours) | 0.86M | 0.25 | 3.28 | **6.68** | **30.57** |
| ResNet56 + HAM (channel, no adaptive) (ours) | **0.85M** | 0.25 | **3.24** | **6.52** | **30.09** |
| ResNet56 + HAM (channel) (ours) | **0.85M** | 0.25 | **3.24** | **6.39** | **29.79** |

**Table 2**
Comparison of various spatial attention modules. The results indicate that HAM module, no matter what kernel size, achieves competitive performance compared to other state-of-the-art attention modules. Moreover, our HAM module with two-dimensional convolution's kernel size of 7 × 7 performs best. Besides, it is important to apply channel and spatial attention simultaneously.

| Description | Parameters | GFLOPs | Memory (M) | CIFAR-10 Error (%) | CIFAR-100 Error (%) |
|---|---|---|---|---|---|
| ResNet56 [12] (baseline) | 0.85M | 0.25 | 3.24 | 7.24 | 31.85 |
| ResNet56 + SE [15] | 0.86M | 0.25 | 3.28 | 6.84 | 30.94 |
| ResNet56 + ECA [31] | 0.85M | 0.25 | 3.24 | 6.55 | 30.45 |
| ResNet56 + CBAM [20] | 0.86M | 0.26 | 3.29 | 6.59 | 30.50 |
| ResNet56 + RCCA (R=2) [19] | 0.87M | 0.26 | 3.32 | 6.39 | 29.71 |
| ResNet56 + HAM (channel)&CBAM (spatial) | 0.85M | 0.26 | 3.25 | 6.49 | 29.79 |
| ResNet56 + HAM (3 × 3) (ours) | 0.85M | 0.26 | 3.25 | 6.49 | 29.74 |
| ResNet56 + HAM (5 × 5) (ours) | 0.85M | 0.26 | 3.25 | 6.34 | 29.68 |
| ResNet56 + HAM (7 × 7) (ours) | **0.85M** | 0.26 | **3.26** | **6.25** | **29.58** |

the separation rate λ to 0.6 in this ablation experiment. Then, we perform average-pooling and max-pooling operations on both of groups to compute two pair of 2*D* descriptors that encode channel information at each pixel over all spatial positions. Next, we concatenate the outputs of pooling operations and send them to a shared convolution layer with kernel size of 7 × 7 to generate a pair of 2*D* attention map. After performing a series of nonlinear operations such as the batch normalization, the ReLU function and the sigmoid function, the final spatial attention maps are given.

In order to evaluate our design philosophy, we set a comparison between 5 methods: SE module [15], ECA module [31], CBAM module [20], RCCA(R=2) module [19] and our HAM module. In addition, we put some effort into searching the effect of a kernel size of the shared two-dimensional convolution. We compare three kinds of kernel sizes of convolution layer: 3, 5 and 7. Due to our final goal is to use both channel and spatial attention module together, we combine the spatial attention with the previously designed channel attention module in the following experiments. Thus, we add another variant module for comparison: the combination of the channel attention submodule in our HAM and the spatial attention submodule in CBAM. For the convenience of marking, the combination is named HAM(channel)&CBAM(spatial).

The results of experiments are shown in Table 2. We can find that our HAM module, no matter what kernel size of the two-dimensional convolution, achieves competitive accuracy performance compared to other attention modules. Comparing the accuracy of HAM and HAM(channel)&CBAM(spatial), we can obviously notice that our spatial attention submodule achieves a higher accuracy. This result of comparison shows that our designed spatial attention submodule has finer attention inference, demonstrating the effectiveness of our proposed channel separation technique. What's more, in the comparison between different convolution kernel sizes, we observe that using larger kernel size can perform better in these cases. This phenomenon indicates that the larger receptive field, the better it is to encode the significant spatial features. Taking this factor into consideration, we decide to adopt the kernel size of 7 × 7 in the convolution layer to compute the spatial attention. Note that our proposed HAM module, which combines channel and spatial attention outperforms SE [15] and ECA [31] modules as they use only channel attention and also outperform the RCCA module [19] as it only uses spatial attention. This shows applying channel and spatial attention simultaneously is necessary. As a brief conclusion, our design method of the spatial attention is effective.

**Separation rate** λ**.** The separation rate λ is a hyperparameter which gives us a boundary between important and sub-important groups of channels. Thus, the separation rate λ may have an influence on the performance of HAM. For exploring a better boundary, we conduct experiments with a range of different λ values.

The experimental results are shown in Table 3. From the table, we can observe that HAM performance is robust to a range of different separation rate. Taking the value of λ closer to 0.5,that is the number of channels in the important group and the sub-important group is equal to half, does not monotonically improve HAM performance. We can set λ = 0.6 to achieve better performance of HAM, which is a good boundary between the important and sub-important groups. However, in practice, taking an identical value of λ throughtout various network models may not be optimal, so that we can fine tune λ to achieve further improvements. For easy comparison, we take λ to be 0.6 in following experiments.

**Table 3**
Comparison of differnet values of λ. Our HAM performance is robust to different separation rate λ while fine tuning the value of it can further improve the accuracy.

| Description | CIFAR-10 Error (%) | CIFAR-100 Error (%) |
|---|---|---|
| ResNet56 [12] (baseline) | 7.24 | 31.85 |
| ResNet56 + HAM (λ=0.8) | 6.63 | 30.58 |
| ResNet56 + HAM (λ=0.7) | 6.41 | 30.37 |
| ResNet56 + HAM (λ=0.6) | **6.25** | **29.58** |
| ResNet56 + HAM (λ=0.5) | 6.38 | 29.84 |

**Final scheme.** Throughout the whole ablation studies, we determine the final channel attention module, the final spatial attention module and the final value of λ. As shown in Fig. 2 and Fig. 3, we use our proposed adaptive mechanism and a fast one-dimensional convolution with kernel size of *k* for the channel attention module; we use our proposed channel separation technique and a shared two-dimensional convolution with kernel size of 7 × 7 for the spatial attention module. Besides, we set the separation rate to 0.6 to achieve a better performance. Finally, the overall structure of our HAM is shown in Fig. 1. Our HAM can be emdedded into any state-of-the-art convolutional architectures to improve the networks performance. As we can see from the following experiments, our attention module achieves competitive results among other state-of-the-art attention modules.

### *4.2. Image classification on CIFAR datasets*

We perform image classification experiments on CIFAR datasets to assess our attention module rigorously. We follow the training rules mentioned in the ablation studies (Section 4.1) and embed our module into different network architectures including ResNet [12], WideResNet [24] and ResNeXt [26]. All models except ResNeXt [26] series are trained with a mini-batch size of 128 on one GPU. For models of ResNeXt series, they are trained with a mini-batch size of 64 on one GPU due to the limitation of GPU's memory.

The experimental results are summarized in Table 4. We can clearly observe that the networks embedded with our HAM module are significantly better than all the corresponding baselines, demonstrating that HAM can indeed improve the representation capability of networks. Moreover, compared with other attention

**Table 4**
Classification results on CIFAR datasets. Error rates are reported. All results are the best of 10 runs.

| Architecture | Parameters | GFLOPs | Memory(M) | CIFAR-10 Error(%) | CIFAR-100 Error(%) |
|---|---|---|---|---|---|
| ResNet20 [12] | 0.26M | 0.08 | 0.99 | 8.54 | 34.52 |
| ResNet20 + SE [15] | 0.27M | 0.08 | 1.03 | 8.31 | 33.88 |
| ResNet20 + ECA [31] | 0.26M | 0.08 | 0.99 | 8.07 | 33.33 |
| ResNet20 + CBAM [20] | 0.27M | 0.08 | 1.04 | 8.23 | 33.45 |
| ResNet20 + RCCA(R=2) [19] | 0.28M | 0.09 | 1.07 | 8.15 | 33.37 |
| ResNet20 + HAM (ours) | **0.26M** | 0.08 | **1.00** | **7.54** | **32.56** |
| ResNet32 | 0.46M | 0.14 | 1.75 | 7.96 | 33.36 |
| ResNet32 + SE | 0.47M | 0.14 | 1.79 | 7.67 | 32.85 |
| ResNet32 + ECA | 0.46M | 0.14 | 1.75 | 7.43 | 31.78 |
| ResNet32 + CBAM | 0.47M | 0.14 | 1.80 | 7.58 | 32.10 |
| ResNet32 + RCCA(R=2) | 0.48M | 0.14 | 1.83 | 7.49 | 31.78 |
| ResNet32 + HAM (ours) | **0.46M** | 0.14 | **1.76** | **7.17** | **31.28** |
| ResNet56 | 0.85M | 0.25 | 3.24 | 7.24 | 31.85 |
| ResNet56 + SE | 0.86M | 0.25 | 3.28 | 6.84 | 30.94 |
| ResNet56 + ECA | 0.85M | 0.25 | 3.24 | 6.55 | 30.45 |
| ResNet56 + CBAM | 0.86M | 0.26 | 3.29 | 6.59 | 30.50 |
| ResNet56 + RCCA(R=2) | 0.87M | 0.26 | 3.32 | 6.39 | 29.71 |
| ResNet56 + HAM (ours) | **0.85M** | 0.26 | **3.26** | **6.25** | **29.58** |
| ResNet110 | 1.72M | 0.51 | 6.56 | 6.91 | 29.78 |
| ResNet110 + SE | 1.74M | 0.51 | 6.64 | 6.46 | 29.56 |
| ResNet110 + ECA | 1.72M | 0.51 | 6.56 | 6.31 | 28.88 |
| ResNet110 + CBAM | 1.74M | 0.52 | 6.65 | 6.44 | 29.19 |
| ResNet110 + RCCA(R=2) | 1.74M | 0.51 | 6.65 | 6.23 | 29.52 |
| ResNet110 + HAM (ours) | **1.73M** | 0.53 | **6.58** | **6.14** | **28.62** |
| WideResNet16-8 [24] | 10.79M | 3.08 | 41.16 | 4.80 | 21.97 |
| WideResNet16-8 + SE | 10.87M | 3.08 | 41.47 | 4.72 | 21.82 |
| WideResNet16-8 + ECA | 10.79M | 3.10 | 41.16 | 4.63 | 21.63 |
| WideResNet16-8 + CBAM | 10.87M | 3.08 | 41.49 | 4.68 | 21.69 |
| WideResNet16-8 + RCCA(R=2) | 12.31M | 3.28 | 46.96 | 4.50 | 21.36 |
| WideResNet16-8 + HAM (ours) | **10.79M** | **3.10** | **41.18** | **4.38** | **21.09** |
| ResNeXt29(8 × 64d) [26] | 33.81M | 9.88 | 128.97 | 4.59 | 21.38 |
| ResNeXt29(8 × 64d) + SE | 34.32M | 9.88 | 130.92 | 4.47 | 21.17 |
| ResNeXt29(8 × 64d) + ECA | 33.81M | 9.96 | 128.97 | 4.26 | 20.64 |
| ResNeXt29(8 × 64d) + CBAM | 34.32M | 9.89 | 130.93 | 4.43 | 21.12 |
| ResNeXt29(8 × 64d) + RCCA(R=2) | 39.86M | 10.66 | 152.05 | 4.19 | 20.43 |
| ResNeXt29(8 × 64d) + HAM (ours) | **33.81M** | **9.97** | **130.94** | **3.97** | **20.23** |

methods mentioned above, the models with HAM achieve the highest accuracy, which indicates that our HAM is more powerful in attention inference. These comparisons imply that our proposed method is powerful, showing the efficacy of *the adaptive mechanism* that forms a complementary relationship between two different pooled features and *a fast one-dimensional convolution* that avoids the channel dimensionality reduction and *channel separation technique* that differentiates between different levels of importance of channels.

### *4.3. Image classification on STL-10 dataset*

The STL-10 dataset is an image classification dataset for developing deep learning algorithms. It is widely used in assessing the performance of networks like CIFAR datasets. The STL-10 dataset contains 13,000 labeled color images with the size of 96 × 96, including 10 classes. Each class in this dataset consists of 1300 images and we can divide these images into training data and test data based on a specific proportion. Generally, we divide the dataset into training data and test data in the ratio of $7:3$ [34,35]. Thus, there are 9100 images for training and 3900 images for test. For training this dataset, we apply the standard data augmentation similar to CIFAR-10: 12 pixels are padded on each side and the padded examples are randomly sampled with a 96 × 96 crop or random horizontal flip. For test, we only use original 96 × 96 images. Since STL-10 dataset has less data than CIFAR datasets, we train the models with mini-batch size of 64 not 128. The other parameters are set the same as CIFAR datasets and we also report the error rates of classification on test data. We divide experiments into two parts. First, we embed our HAM into various backbone models such as ResNet [12], WideResNet [24] and ResNeXt [26]. Then, we attempt to increase the depth of ResNet to observe the stability of our HAM. The experimental details are described below.

**Comparison of different backbone models.** We embed our HAM into ResNet [12], WideResNet [24] and ResNeXt [26] and compare the performance with other attention modules mentioned above. We report error rates and all results are the best of 10 runs.

The experimental results are recorded in Table 5. We can clearly see that our HAM module improves the performance of baseline networks significantly, proving that HAM module is robust to different classification datasets in accuracy. Moreover, our proposed HAM module achieves better performance than other comparison attention modules. In ResNet [12] model, HAM achieves 11.21% in classify error rate which has a promotion of 11.0% compared to ECA module [31] and 16.9% compared to CBAM module [20]. The results in WideResNet [24] and ResNeXt [26] show the same trends. In brief, our HAM outperforms other aforementioned attention modules, which demonstrates the effectiveness of HAM. In addition, HAM has less amount parameters than other attention modules as illustrated in Table 5, showing that it is a really lightweight module.

**Comparison of different depth ResNet.** For STL-10 dataset, it has far less data than CIFAR datasets. Thus, the model would be overfitting on training set, resulting in poor performance on test set. In this part, we try to deepen the ResNet [12] models to validate the stability of our HAM. Our focus is on whether the attention module still performs robustly in deep networks with few samples of datasets, but not on pushing the state-of-the-art results,

**Table 5**
Comparison of different CNN architectures on STL-10 dataset. We embed HAM into various backbone models and report error rates. Our HAM achieve the best performance among these state-of-the-art attention modules.

| Architecture | Parameters | GFLOPs | Memory(M) | STL-10 Error(%) |
|---|---|---|---|---|
| ResNet18 [12] | 11.00M | 2.53 | 41.96 | 15.10 |
| ResNet18 + SE [15] | 11.09M | 2.53 | 42.30 | 14.46 |
| ResNet18 + ECA [31] | 11.01M | 2.55 | 42.00 | 12.59 |
| ResNet18 + CBAM [20] | 11.09M | 2.53 | 42.31 | 13.59 |
| ResNet18 + RCCA(R=2) [19] | 12.52M | 2.64 | 47.76 | 12.10 |
| ResNet18 + HAM (ours) | **11.01M** | **2.55** | **42.01** | **11.21** |
| WideResNet18-2 [24] | 43.98M | 10.03 | 167.77 | 14.49 |
| WideResNet18-2 + SE | 44.32M | 10.03 | 169.07 | 14.07 |
| WideResNet18-2 + ECA | 44.00M | 10.09 | 167.85 | 12.54 |
| WideResNet18-2 + CBAM | 44.33M | 10.04 | 169.11 | 13.79 |
| WideResNet18-2 + RCCA(R=2) | 50.03M | 10.48 | 190.85 | 11.74 |
| WideResNet18-2 + HAM (ours) | **44.00M** | **10.10** | **167.86** | **10.90** |
| ResNeXt50(32 × 4d) [26] | 21.14M | 4.93 | 80.64 | 16.44 |
| ResNeXt50(32 × 4d) + SE | 23.65M | 4.94 | 90.22 | 15.68 |
| ResNeXt50(32 × 4d) + ECA | 21.22M | 5.44 | 80.95 | 13.27 |
| ResNeXt50(32 × 4d) + CBAM | 23.66M | 4.96 | 90.26 | 14.82 |
| ResNeXt50(32 × 4d) + RCCA(R=2) | 45.40M | 6.72 | 173.19 | 11.96 |
| ResNeXt50(32 × 4d) + HAM (ours) | **21.23M** | **5.46** | **80.99** | **11.13** |

**Table 6**
Comparison of different depth ResNet on STL-10 dataset. We use ResNet with 18, 34, 50 and 101 layers as the backbone network respectively to observe the impact of various attention modules on deep networks. The validation error rates are reported on the table.

| Architecture | Parameters | GFLOPs | Memory(M) | STL-10 Error(%) |
|---|---|---|---|---|
| ResNet18 [12] | 11.00M | 2.53 | 41.96 | 15.10 |
| ResNet18 + SE [15] | 11.09M | 2.53 | 42.30 | 14.46 |
| ResNet18 + ECA [31] | 11.01M | 2.55 | 42.00 | 12.59 |
| ResNet18 + CBAM [20] | 11.09M | 2.53 | 42.31 | 13.59 |
| ResNet18 + RCCA(R=2) [19] | 12.52M | 2.64 | 47.76 | 12.10 |
| ResNet18 + HAM (ours) | **11.01M** | **2.55** | **42.01** | **11.21** |
| ResNet34 | 21.11M | 5.27 | 80.53 | 14.74 |
| ResNet34 + SE | 21.27M | 5.27 | 81.14 | 15.77 |
| ResNet34 + ECA | 21.13M | 5.30 | 80.60 | 15.26 |
| ResNet34 + CBAM | 21.27M | 5.28 | 81.16 | 13.10 |
| ResNet34 + RCCA(R=2) | 22.63M | 5.38 | 86.33 | **11.95** |
| ResNet34 + HAM (ours) | **21.13M** | **5.31** | **80.61** | **12.44** |
| ResNet50 | 20.73M | 5.06 | 79.08 | 15.23 |
| ResNet50 + SE | 23.25M | 5.08 | 88.69 | 15.26 |
| ResNet50 + ECA | 20.82M | 5.57 | 79.42 | 16.70 |
| ResNet50 + CBAM | 23.25M | 5.09 | 88.69 | 14.38 |
| ResNet50 + RCCA(R=2) | 44.99M | 6.85 | 171.62 | 13.51 |
| ResNet50 + HAM (ours) | **20.82M** | **5.60** | **79.43** | **13.13** |
| ResNet101 | 39.72M | 10.56 | 151.52 | 17.46 |
| ResNet101 + SE | 44.47M | 10.59 | 169.64 | 18.49 |
| ResNet101 + ECA | 39.90M | 11.43 | 152.21 | 17.90 |
| ResNet101 + CBAM | 44.47M | 10.62 | 169.66 | 18.08 |
| ResNet101 + RCCA(R=2) | 63.98M | 12.35 | 244.06 | **12.49** |
| ResNet101 + HAM (ours) | **39.90M** | **11.48** | **152.22** | **14.05** |

so we abide by the experimental settings mentioned above which have no bells and whistles.

Table 6 summarizes the experimental results. We can clearly see that a decreasing trend in the performance of networks on the test set as the networks go deeper. The main reason for this phenomenon is the small number of data in STL-10 dataset, which makes deeper networks are overfitting on training set. Table 6 obviously shows that our HAM only loses about 3% accuracy of its performance from 18 to 101 layers while SE [15], ECA [31] and CBAM [20] modules lose about 5% accuracy. However, CCNet [19] loses about 1.5% accuracy, which is less than HAM. In addition, comparing the performance of 101 layers HAM-integrated network (ResNet101+HAM) with ResNet18 [12], we can clearly observe that our HAM still achieves better performance. This demonstrates that our HAM has stability when the dataset has less data. In other words, HAM can reduce the negative impact of less data on deeper networks performance. This shows a great development potential of HAM.

### *4.4. Attention mechanism visualization with Grad-CAM*

In order to reflect the superiority of HAM more directly, we use the Grad-CAM [36] to analyze its function in networks. Grad-CAM is an effective visualization method like a heat map which can explain how the CNNs make decisions in classification by using gradients to compute the important regions of convolution layers. The gradients are calculated according to a specific class, so that we can clearly see the attended regions which have a positive impact on the prediction of this category. By observing the results of Grad-CAM, we can know the networks focus on 'what' and 'where', that is attention mechanism. We compare the visualization results of ResNet18+HAM with ResNet18 and ResNet18+CBAM [20] based on the STL-10 dataset. Fig. 5 shows the visualization results.

As shown in Fig. 5, we can observe that the Grad-CAM visualization result of HAM-integrated network outperform other networks in covering the target category regions. This demonstrates

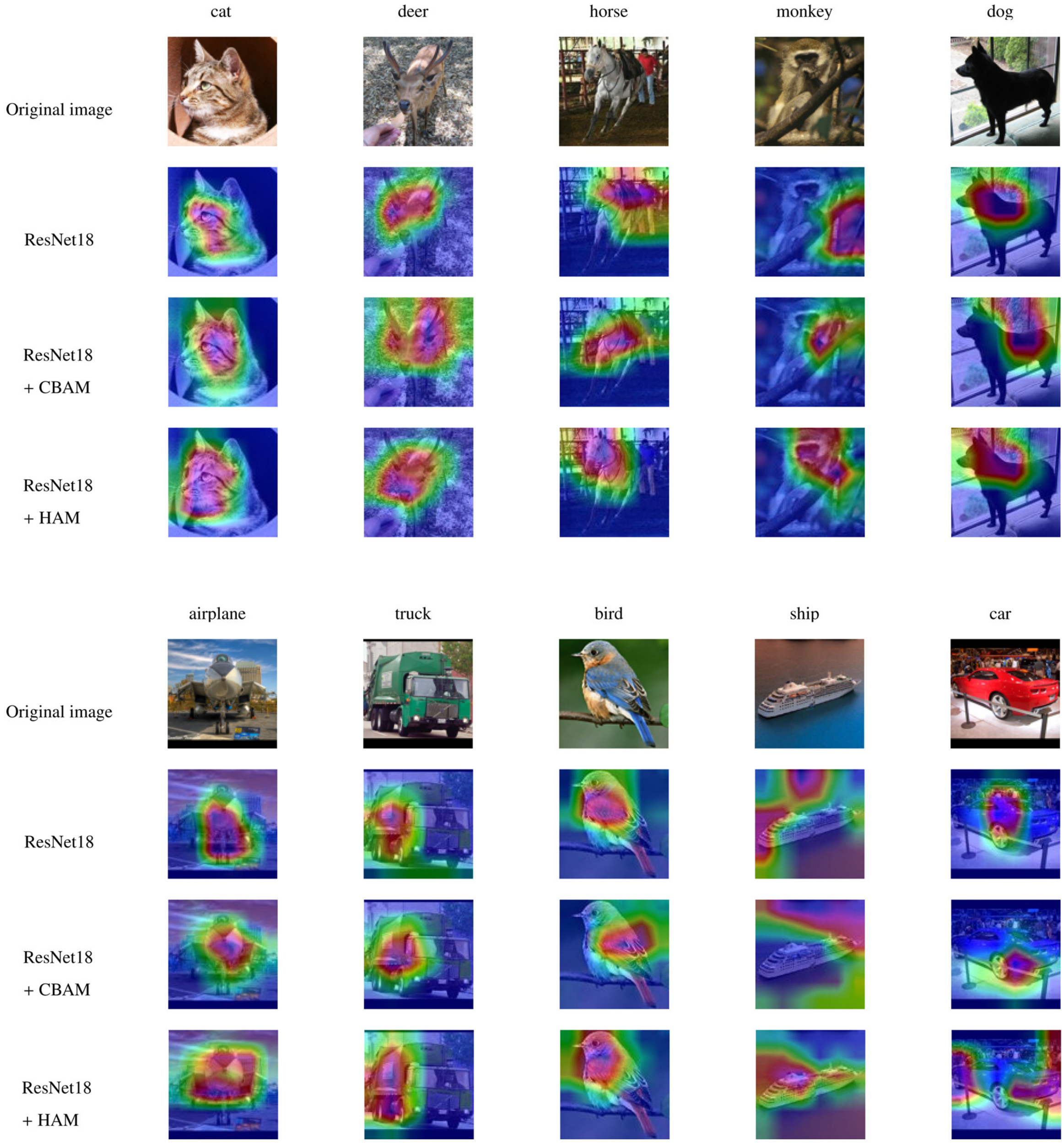

**Fig. 5.** The visualization results by Grad-CAM. We compare three visualization results: ResNet18 [12] (baseline), ResNet18+CBAM [20] and ResNet18+HAM in the figure. We show the ground-truth label on the top of each original image. The results clearly show us that our HAM can better focus on important regions.

that the network with HAM can really learn to concentrate on target class regions. Moreover, we can speculate that HAM module can lead the networks to pay more attention to important features while ignoring the unimportant ones.

## 5. Conclusion

We have proposed a new lightweight and powerful attention module, Hybrid Attention Module (HAM), which improves the representation capability of CNNs. HAM consists of two distinctive submodules, channel and spatial attention module. For channel attention, we suggest using our designed adaptive mechanism and a fast one-dimensional convolution, which would produce finer channel attention than other channel attention modules. For spatial attention, we apply our proposed channel separation technique along the channel axis. After the channel separation operation, we use a shared two-dimensional convolution layer to generate a pair of 2*D* spatial attention maps. This would further push the networks performance. Besides, through ablation experiments, we conclude that spatial attention needs a larger receptive field so that we adopt the two-dimensional convolution with kernel size of 7 × 7. At last, we have determined the value of separation rate $\lambda$, that is 0.6. Our final attention module (HAM) can effectively learn to focus on target class regions.

To evaluate HAM, we first conducted ablation experiments on CIFAR datasets which show the validity of our design schemes of the channel and spatial attention. Then, we compared HAM with other state-of-the-art attention modules including SE [15], ECA [31] and CBAM [20] by embedding them into different CNNs on CIFAR and STL-10 datasets. The experimental results show that HAM is powerful in attention inference and is robust to different classification datasets. Finally, on the STL-10 dataset, we visualize ‘what’ and ‘where’ do HAM-integrated networks focus on by using the Grad-CAM. The visualization results demonstrate that HAM can really induce the networks pay more attention on target object regions, which is the true meaning of attention.

Note that the spatial attention submodule of our HAM needs larger receptive field. As a result, we will develop a new approach to obtain the full-image contextual information instead of applying a convolution layer in the next work. In addition, our HAM still performs robustly in deep models with small datasets, which leads us to argue that the attention mechanism can really understand knowledge of the target class. This conjecture inspires our next work, applying the attention mechanism to the class-incremental learning. In the future, we will develop self-organized and incremental mechanisms for HAM in order to apply it in class-incremental learning. What's more, we will further investigate the few-shot class-incremental learning based on HAM module. We hope our HAM can be applied in various computer vision tasks as an important component.

## Declaration of Competing Interest

The authors declared that they have no conflicts of interest in this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

## Acknowledgements

This work was supported in part by the Natural Science Foundation of Hebei Province of China (Grant No. F2020203003), in part by Zhejiang Provincial Natural Science Foundation (LR19F020005).

## References

[1] L. Li, L. Ma, L. Jiao, F. Liu, Q. Sun, J. Zhao, Complex contourlet-CNN for polarimetric sar image classification, Pattern Recognit 100 (2020) 107110, doi:10.1016/j.patcog.2019.107110.

[2] N. Tong, Y. Tang, B. Chen, L. Xiong, Representation learning using attention network and cnn for heterogeneous networks, Expert Syst Appl 185 (2021) 115628, doi:10.1016/j.eswa.2021.115628.

[3] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, doi:10.1109/5.726791.

[4] J. Yuan, H.-C. Xiong, Y. Xiao, W. Guan, M. Wang, R. Hong, Z.-Y. Li, Gated CNN: integrating multi-scale feature layers for object detection, Pattern Recognit 105 (2020) 107131, doi:10.1016/j.patcog.2019.107131.

[5] X. Yang, T. Wu, N. Wang, Y. Huang, B. Song, X. Gao, Hcnn-psi: a hybrid CNN with partial semantic information for space target recognition, Pattern Recognit 108 (2020) 107531, doi:10.1016/j.patcog.2020.107531.

[6] Z. Dong, C. Jing, M. Pei, Y. Jia, Deep cnn based binary hash video representations for face retrieval, Pattern Recognit 81 (2018) 357–369, doi:10.1016/j.patcog.2018.04.014.

[7] Q. Zhang, C. Bai, Z. Liu, L.T. Yang, H. Yu, J. Zhao, H. Yuan, A gpu-based residual network for medical image classification in smart medicine, Inf Sci (Ny) 536 (2020) 91–100, doi:10.1016/j.ins.2020.05.013.

[8] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: 2012 Neural Information Processing Systems(NIPS), volume 25, 2012, pp. 1097–1105, doi:10.1145/3065386.

[9] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015 International conference on learning representations (ICLR) abs/1409.1556 (2015).

[10] M. Lin, Q. Chen, S. Yan, Network In Network, in: 2014 Proceedings of the IEEE International Conference on Learning Representations(ICLR), 2014.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, doi:10.1109/CVPR.2015.7298594.

[12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 7, 2015.

[13] Y. Fan, J. Liu, R. Yao, X. Yuan, Covid-19 detection from x-ray images using multi-kernel-size spatial-channel attention network, Pattern Recognit 119 (2021) 108055, doi:10.1016/j.patcog.2021.108055.

[14] C. Pu, H. Huang, L. Yang, An attention-driven convolutional neural network-based multi-level spectral-spatial feature learning for hyperspectral image classification, Expert Syst Appl (2021) 115663, doi:10.1016/j.eswa.2021.115663.

[15] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans Pattern Anal Mach Intell 42 (8) (2020) 2011–2023, doi:10.1109/TPAMI.2019.2913372.

[16] Y. Chen, Y. Kalantidis, J. Li, S. Yan, J. Feng, $A^2$-nets: Double attention networks, in: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

[17] Z. Gao, J. Xie, Q. Wang, P. Li, Global second-order pooling neural networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[18] H. Ma, G. Han, L. Peng, L. Zhu, J. Shu, Rock thin sections identification based on improved squeeze-and-excitation networks model, Computers & Geosciences 152 (2021) 104780, doi:10.1016/j.cageo.2021.104780.

[19] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, T.S. Huang, Ccnet: criss-cross attention for semantic segmentation, IEEE Trans Pattern Anal Mach Intell (2020), doi:10.1109/TPAMI.2020.3007032. 1–1

[20] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional Block Attention Module, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 3–19.

[21] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, C.-W. Lin, Deep learning on image denoising: an overview, Neural Networks 131 (2020) 251–275, doi:10.1016/j.neunet.2020.07.025.

[22] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 630–645.

[23] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: 2017 Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, in: AAAI'17, AAAI Press, 2017, pp. 4278–4284.

[24] S. Zagoruyko, N. Komodakis, Wide residual networks, in: E.R.H. Richard C. Wilson, W.A.P. Smith (Eds.), Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, 2016, pp. 87.1–87.12, doi:10.5244/C.30.87.

[25] D. Han, J. Kim, J. Kim, Deep pyramidal residual networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6307–6315, doi:10.1109/CVPR.2017.668.

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5987–5995, doi:10.1109/CVPR.2017.634.

[27] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269, doi:10.1109/CVPR.2017.243.

[28] X. Xing, Y. Yuan, M.Q.-H. Meng, Zoom in lesions for better diagnosis: attention guided deformation network for wce image classification, IEEE Trans Med Imaging 39 (12) (2020) 4047–4059, doi:10.1109/TMI.2020.3010102.

[29] M. Zhu, L. Jiao, F. Liu, S. Yang, J. Wang, Residual spectral–spatial attention network for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 59 (1) (2021) 449–462, doi:10.1109/TGRS.2020.2994057.

[30] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6450–6458, doi:10.1109/CVPR.2017.683.

[31] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, Eca-net: Efficient channel attention for deep convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2921–2929, doi:10.1109/CVPR.2016.319.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, Tensorflow: A system for large-scale machine learning, in: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, in: OSDI'16, USENIX Association, USA, 2016, pp. 265–283.

[34] M.W. Browne, Cross-validation methods, J Math Psychol 44 (1) (2000) 108–132, doi:10.1006/jmps.1999.1279.

[35] F. Maleki, N. Muthukrishnan, K. Ovens, C. Reinhold, R. Forghani, Machine learning algorithm validation: from essentials to advanced applications and implications for regulatory certification and deployment, Neuroimaging Clin. N. Am. 30 (4) (2020) 433–445, doi:10.1016/j.nic.2020.08.004. Machine Learning and Other Artificial Intelligence Applications

[36] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626, doi:10.1109/ICCV.2017.74.

**Guoqiang Li** received his Ph.D. degree from Yanshan University in 2013. He is currently working in the School of Electrical Engineering, Yanshan University, Qinhuangdao, China. His research interests include thermal process automation, computer vision, machine learning, artificial intelligence control and heuristic optimization algorithm.

**Qi Fang** is working on his M.S degree in the School of Electrical Engineering, Yanshan University, Qinhuangdao, China. He received the B.S degree from School of Mechanical and Electrical Engineering, Xi'an Polytechnic University, Xi'an, China, in 2018. His current research interests include deep learning and computer vision.

**Linlin Zha** is working on her Ph.D. degree in the School of Electrical Engineering, Yanshan University, Qinhuangdao, China. She received the B.S degree from Polytechnic College of Hebei University of Science and Technology, Shijiazhuang, China, in 2015. She received the M.S degree from Yanshan University, Qinhuangdao, China, in 2018. Her current research interests include machine learning and neural network.

**Xin Gao** received his M.S degree from the School of Electrical Engineering, Yanshan University, Qinhuangdao, China, in 2020. He received the B.S degree from School of Mechanical and Electrical Engineering, Daqing Normal University, Daqing, China, in 2018. His current research interests include deep learning and computer vision.

**Nenggan Zheng** received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2009. He is currently an Associate Professor in computer science with the Academy for Advanced Studies, Zhejiang University. His current research interests include artificial intelligence, embedded systems, and brain computer interface.