Received: 8 October 2021 | Revised: 29 September 2022 | Accepted: 5 December 2022

IET Computer Vision

DOI: 10.1049/cvi2.12166

ORIGINAL RESEARCH

# CAT: Learning to collaborate channel and spatial attention from multi-information fusion

Zizhang Wu[1] | Man Wang[1] | Weiwei Sun[1] | Yuchen Li[1] | Tianhao Xu[1,2] | Fan Wang[1] | Keke Huang[3]

[1]Zongmu Technology, Shanghai, China

[2]Technical University of Braunschweig, Braunschweig, Germany

[3]Central South University, Changsha, China

**Correspondence**

Tianhao Xu.
Email: xutianhao2018@gmail.com

**Funding information**

Open Access funding enabled and organized by Projekt DEAL.

[Correction added on 28 February 2023, after first online publication: The corresponding author was changed from Zizhang Wu to Tianhao Xu.]

**Abstract**

Channel and spatial attention mechanisms have proven to provide an evident performance boost of deep convolution neural networks. Most existing methods focus on one or run them parallel (series), neglecting the collaboration between the two attentions. In order to better establish the feature interaction between the two types of attentions, a plug-and-play attention module is proposed, which is termed as 'CAT'—activating the Collaboration between spatial and channel Attentions based on learned Traits. Specifically, traits are represented as trainable coefficients (i.e. colla-factors) to adaptively combine contributions of different attention modules to fit different image hierarchies and tasks better. Moreover, the global entropy pooling is proposed apart from global average pooling and global maximum pooling (GMP) operators, which is an effective component in suppressing noise signals by measuring the information disorder of feature maps. A three-way pooling operation is introduced into attention modules and the adaptive mechanism is applied to fuse their outcomes. Extensive experiments on MS COCO, Pascal-VOC, Cifar-100, and ImageNet show that our CAT outperforms the existing state-of-the-art attention mechanisms in object detection, instance segmentation, and image classification. The model and code will be released soon.

**KEYWORDS**

channel attention, dynamic learning, entropy pooling, spatial attention

## 1 | INTRODUCTION

Convolutional neural networks (CNNs) are one of the most powerful learning algorithms for vision tasks and have shown exemplary performance in areas like image classification, object detection, and semantic segmentation [1–5]. The general form of CNNs usually consists of multiple stages dealing with hierarchical features, exploiting spatial or other correlations in data at a multi-level. There are three primary factors during the learning process: sparse interaction, parameter sharing, and equivariant representation.

From the three perspectives mentioned above, various adjustments and improvements were performed on CNN to deal with more complex and heterogeneous problems. CNN-based methods became extremely prevalent after the great success of AlexNet [6]. After that, innovations in CNN components ushered in rapid development. The split, transform, and merge became a routine process, allowing the abstraction of features at different spatial scales. Afterwards, the concept was applied to most succeeding methods, including the attention mechanisms.

The performance boost of CNN mainly lies in adjustments on network structure [7, 8] and introductions of effective normalisation and pooling mechanisms [9–11]. Attention mechanism works on assigning extra weights to significant factors through applying residual connections on CNN backbones. Recent attention-based networks [12–14] have made preferable improvements in tasks of classification, segmentation, and detection etc.

Squeeze-and-Excitation Networks (SENet) [12], the most representative attention-based network, designs a squeeze-and-

excitation module that extracts channel-wise weights by applying global average pooling (GAP) operators and assigns them to each spatial plane of a feature map. Such architecture brings notable performance gain compared to baselines and costs little extra computational capacity. ECANet (Efficient Channel Attention) [14] designs a 1-D convolution kernel to conduct the interaction among channels so that it further reduces the computational complexity. Later studies [15–18] that work on developing techniques of squeeze and excitation only achieve limited progress due to the lack of information about spatial-wise interaction [19–21]. To tackle this problem, the authors in ref. [22–24] suggest architectures to take both spatial and channel interactions into consideration. Particularly, Convolutional Block Attention Module (CBAM) [24] achieves notable improvements than SENet by designing a spatial-wise attention module and sequentially connecting it with the channel-wise attention module.

Most previous attention-based methods only focus on the channel or spatial attention or embed them through direct addition or concatenation [22, 24, 25]. Given such a situation, an adaptive mechanism could benefit the fusion of different attentions more feasibly. In other words, our framework Collaborates channel and spatial Attention contributions based on their Traits (CAT) in a multi-information-fusion style. We also notice that the widely used GAP [9] and GMP [26] operators have disadvantages in capturing rich texture distribution information of large ranges. Moreover, they probably cause excessive background noise as the former operator treats the background and object regions equally and the latter one is merely susceptible to extreme features [27]. Hence, we introduce global entropy pooling (GEP) operators into our network, which measure spatial and channel systems' information disorders by calculating the entropy of input feature maps (Section 3). Our ablation experiments illustrate that placing these attention operators in parallel and fusing their outcomes with the adaptive mechanism enhance the performance of our framework in segmentation and detection tasks etc.

The overview of the CAT framework is specifically present in Figure 1. We disentangle channel and spatial attention modules and assign two exterior colla-factors $C_w$ and $S_w$ to them to attain the coordination and cooperation of the two modules. It is noteworthy that we can embed the CAT in all appropriate positions in networks, such as the residual operation of each block in ResNet. In each attention module, we use the GEP, GMP, and GAP to extract three raw attention maps from input features $F$ for channel and spatial attentions in $\mathbb{R}^{1\times1\times C}$ and $\mathbb{R}^{H\times W\times1}$. Inside the channel attention module, to effectively combine the outputs of the above operators, we adopt an element-wise addition to equilibrate their effects and design a shared multilayer perception (MLP) structure to capture their cross-channel interactions. On the other hand, inside the spatial attention module, we construct a system consisting of a dynamic and linearly weighted fusion operator and a $7 \times 7$ convolution to capture the spatial neighbourhood information.

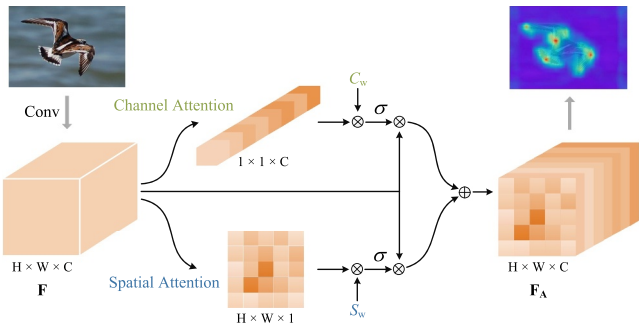



**FIGURE 1** Our CAT framework—Channel and spatial attention collaborate via the linear combination, which is controlled by the exterior colla-factors (i.e. $C_w$ and $S_w$). We set the colla-factors as trainable parameters in network. Additionally, to generate attention, we fuse information collected from three pooling methods, that is, global max pooling (GMP), global average pooling (GAP), and our proposed global entropy pooling (GEP). $\otimes$ and $\oplus$ denote element-wise multiplication and element-wise addition. $\sigma$ is a sigmoid function. Best view in colour and zoom in.

We verify the performance of CAT in object detection, instance segmentation, and image classification on Pascal-VOC, MS COCO, and Cifar-100 datasets, respectively. Extensive experiments show that the proposed CAT outperforms state-of-the-art attention mechanisms such as SENet [12], CBAM [24], and ECANet [14].

Our contributions can be summarised as follows: (1) We propose a collaborative attention framework by dynamically learning the interaction between the channel and spatial attention modules so that our framework is adaptive to different embedded image hierarchies and tasks; (2) We propose the GEP attention operator and design, an adaptive mechanism to capture the inherent collaboration relationship of different attention operators (GEP, GAP, and GMP), which increases the texture sensitivity of our framework while it provides negligible extra parameters; (3) We show our network outperforms previous attention-based architectures with no large extra computational complexity required. Furthermore, it reveals the potential of elaborately designed attention extraction structures that selectively emphasise the interest factors in a wide range of computer vision tasks.

## 2 | RELATED WORK

### 2.1 | Attention mechanism

Many recent research studies focus on applying attention mechanisms in a series of computer vision tasks [12, 14, 26]. The most representative attention-based network, SENet [12], employs GAP to obtain weights of attention for spatial planes and conducts cross-channel interaction by using fully connected layers. Later arts like GENet (Gather-excite) [15], SGENet (Spatial group-wise enhance) [28], GCNet [26], and SKNet [29] work on improving the SENet (Selective kernel network) [12] in its mechanism of extracting attention weights and its fusing method of combining attention module outputs

and feature maps. However, the employment of fully connected layers in these works for conducting information interaction among channels leads to great computational complexity. To tackle this problem, ECANet [14] employs a 1-D convolution kernel, instead, that boosts the training and implementation speed with competitive network performance.

Other research studies start to present attention modules in the spatial aspect [19–21, 30] to further improve the performance of neural networks. Spatial Transformer Network [19] pays additional attention to regions of interest and transforms them into expected postures to facilitate the learning of backbone and output layers. Wang H et al. [30] use the summation operator to replace GAP for alleviating the loss of information caused in person-type object re-identification. For capturing long-range dependencies, Wang X et al. [13] propose a non-local method to measure relationships of all possible couples for whole sets of pixels in feature maps. To further combine channel and spatial attention contributions, CBAM [24] sequentially introduces both of them into attention modules in which GAP and GMP act simultaneously to extract attention weights. Networks of these studies [23, 25] adopt a similar non-local way to deal with spatial attention and combine the spatial and channel outcomes in parallel structure for segmentation tasks.

Along with the CBAM, Bottleneck Attention Module (BAM [31]) also infers the attention map along two separate pathways, channel and spatial, but constructs hierarchical attention at bottlenecks. Recently, FacNet [32] considers the attention mechanism by regarding the channel representation problem as a compression process using frequency analysis, which proved that GAP is a special case of the feature decomposition in the frequency domain. CAEM [33] proposed an attention mechanism by embedding positional information into the channel attention, which is the coordinate attention. There are also works in low-level vision fields that explored the attention mechanism. CycleISP [34] and MIR-Net [35] achieved better results with attention mechanisms in RAW and sRGB image denoising, colour matching, image restoration, and enhancement tasks. Despite the impact of above methods on CNNs, they neglect the important difference between the channel-wise and spatial-wise attention contributions, which is important when dealing with various image hierarchies and tasks. We thus propose a framework to dynamically learn collaborating exterior and interior interactions that are between and within different attention modules, respectively.

## 2.2 | Information entropy

The objective of measuring information entropy is to count the information uncertainty of a system [36–38]. We expect a system to have high uncertainty if its predictions are of large entropy and vice versa [27]. The Entropy Guided Adversarial model proposed in ref. [27] introduces an entropy loss function for guiding the CNN to make pixel-level detections of objects. Some other methods also employ entropies for information pruning [36, 37, 39]. Ref. [36] combines impacts of kernel sparsity and entropy to quantify the importance of a feature map for the task of model compression. Li Y et al. [37] also constructs a quantitative model by defining a weighted entropy to comprehensively measure the importance and frequency of filters.

There are also successful applications of entropy in semi-supervised and inter-domain adaptive learnings [38, 40–42]. For example, Saito K et al. [38] employ the minimax entropy (MME) method to train its classifier by reducing differences among distributions and learning discriminative features in tasks. The utiliation of entropy minimisation [40] can train the classifier from unmarked or partially marked data. Moreover, Wan W et al. [43] proposed information entropy-based feature pooling but only for classic CNNs. Inspired by above applications of entropy, we design and introduce a GEP operator for our attention-based framework. The proposed GEP is able to complement GAP and GMP for restraining the effect of irrelevant background noises during pooling. These pooling operators help to capture attention maps in three different formats and information extraction views to improve attention modules' sensitivity and robustness.

# 3 | PROPOSED METHOD

In this section, we will introduce the proposed CAT framework in detail which focuses on dynamically learning the collaborative relationship between spatial and channel attention modules. In each attention module, the GEP complements GMP and GAP to extract attention weights. Hence, we apply the adaptive mechanism that firstly captures the collaborative relationship of above pooling operators based on their traits and then fuses spatial and channel attention modules for measuring their interactions.

## 3.1 | Overview

The overall structure of the proposed CAT is shown in Figure 1. We linearly combine the collaborative relationship between spatial and channel attention modules since they are sensitive to different network depths and tasks. We introduce exterior colla-factors $C_w$ and $S_w$ for channel and spatial attention modules and multiply them with corresponding attention weights. We then multiply the modified weights with input feature map $F$ to generate two attention maps. Eventually, by applying the element-wise addition operation, we can obtain the final feature map $F_A$.

$$F_{\mathrm{A}} = \left(F \otimes \sigma\left(C'_{\mathrm{A}} C_{\mathrm{w}}\right)\right) + \left(F \otimes \sigma\left(S'_{\mathrm{A}} S_{\mathrm{w}}\right)\right) \quad (1)$$

where $C'_{\mathrm{A}}$ and $S'_{\mathrm{A}}$ represent raw attention maps of the channel and spatial dimensions. The framework initialises learnable $C_w$ and $S_w$ as 0 and applies a softmax function to normalise them. $\otimes$ denotes an element-wise multiplication.

## 3.2 | Channel attention module

$F \in \mathbb{R}^{H\times W\times C}$ is an input feature map, where $W$, $H$, and $C$ are width, height, and channel dimensions. As shown in Figure 2a, for the channel attention module, we adopt the GAP, GMP, and GEP to extract three-dimensional attention weights whose dimensions are $\mathbb{R}^{1\times1\times C}$ as follows:

$$C_{\text{Avg}}; C_{\text{Max}}; C_{\text{Ent}} = MLP\left(C'_{\text{Avg}}; C'_{\text{Max}}; C'_{\text{Ent}}\right) \tag{2}$$

where $C'_{\text{Avg}}$, $C'_{\text{Max}}$, and $C'_{\text{Ent}}$ denote the GAP, GMP, and GEP attention maps. MLP is a parameter shared network with two fully connected (FC) layers and an activation layer. We use the first FC to reduce channel dimension into $\mathbb{R}^{1\times1\times\frac{C}{r}}$, where $r = 16$ is the reduction ratio. The following ReLU function activates the former output. Hence, we adopt the last FC to introduce non-linearity and extend channel dimension into $\mathbb{R}^{1\times1\times C}$.

In order to pay more attention to channels with rich textures, we employ GEP to extract the corresponding attention weight of each channel. Here, the GEP is defined as follows:

$$C'_{\text{Ent}} = -\sum_{i=1}^{H\times W} p_i log p_i \tag{3}$$

where $p_i = \text{softmax}\left(F_i^{1\times1\times C}\right)$ is the probability of location $i$ in each feature map. We normalise $C'_{\text{Ent}}$ in the interval [−1, 1] before inputting it into MLP through the function:

$$\chi_i \to \frac{\chi_i - \chi_{\min}}{\chi_{\max} - \chi_{\min}} \tag{4}$$

where $\chi_i$, $\chi_{\min}$, and $\chi_{\max}$ are for the operation on the $C'_{\text{Ent}}$.

As shown in Figure 2a, the element-wise addition can obtain the channel attention map $C'_{\text{A}}$. To aggregate the above GAP, GMP, and GEP attention operators adaptively, the interior colla-factors $C_\alpha$, $C_\beta$, and $C_\gamma$ collaborate with the operators as follows:

$$C'_{\text{A}} = C_{\text{Avg}}C_\alpha + C_{\text{Max}}C_\beta + C_{\text{Ent}}C_\gamma \tag{5}$$

where $C_\alpha$, $C_\beta$, and $C_\gamma$ are initialised as 0 and learnable through training. GAP and GMP retain the global average information and the most distinctive information of the input $F$. Extra noises are easy to disturb the distinctive information from GMP. Therefore, we employ a 1-D Gaussian low-pass filter ($k \times 1$) before all GMP operations to alleviate the inference that causes adverse effects for obtaining more effective maximum response characteristics. Furthermore, $k$ is set to 5 in this study after comparisons in experiments.

## 3.3 | Spatial attention module

As shown in Figure 2b, similar to the channel attention module, the spatial attention module also uses the GAP, GMP, and GEP to obtain three attention maps in $\mathbb{R}^{H\times W\times1}$. We encode channel information at each pixel over all spatial locations and output one feature map.

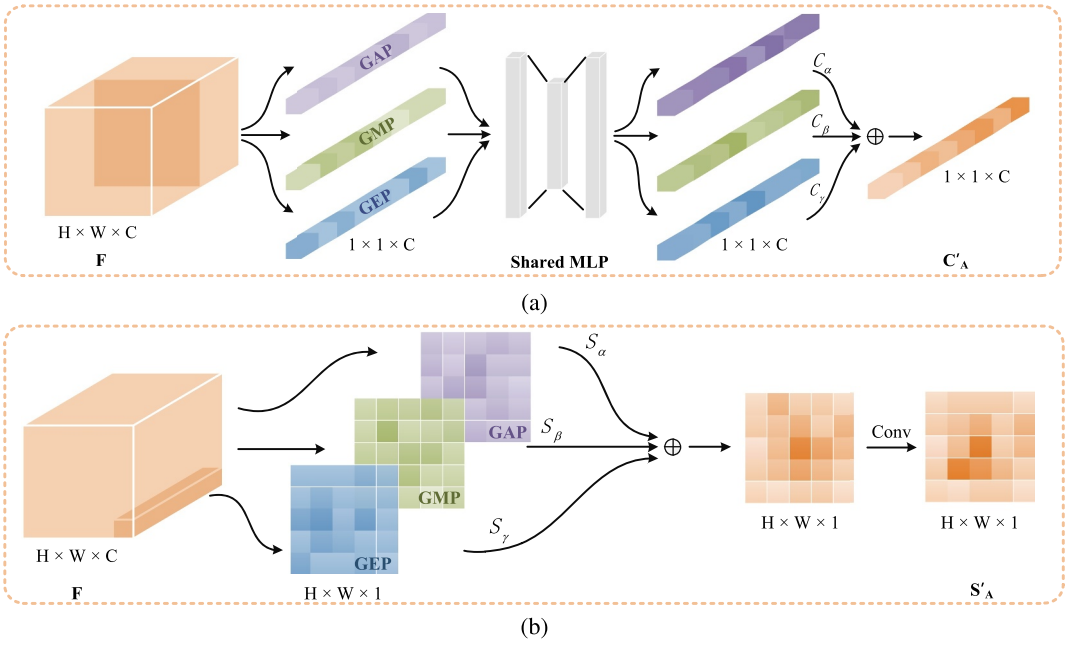



**FIGURE 2** Illustrations of the attention modules: (a) Channel attention module. We employ channel global average pooling (GAP), GMP, and global entropy pooling (GEP) to extract raw attention maps. A shared multilayer perception (MLP) and an element-wise addition are used to combine the outputs. Three interior colla-factors $C_\alpha$, $C_\beta$, and $C_\gamma$ are, respectively, assigned to GAP, GMP, and GEP to collaborate with them jointly and dynamically. (b) Spatial attention module. It gathers outputs of spatial GAP, GMP, and GEP attention maps. Three interior colla-factors $S_\alpha$, $S_\beta$, and $S_\gamma$ are, respectively, assigned to GAP, GMP, and GEP to collaborate with them jointly and dynamically. Best view in colour and zoom in.

In spatial attention module, the GEP is defined as follows:

$$S_{\text{Ent}} = -\sum_{j=1}^{C} p_j log p_j \tag{6}$$

$p_j = \text{softmax}\left(F_j^{H\times W\times 1}\right)$ is the probability for a channel.

To aggregate the above GAP, GMP, and GEP pooling operators adaptively, we introduce interior colla-factors $S_\alpha$, $S_\beta$, and $S_\gamma$ to collaborate with the operators. We then multiply them with corresponding feature maps as follows:

$$S'_A = \sigma\left(\text{Conv}^{7\times 7}\left(-S_{\text{Avg}}S_\alpha + S_{\text{Max}}S_\beta + S_{\text{Ent}}S_\gamma\right)\right) \tag{7}$$

where $S_{\text{Avg}}$, $S_{\text{Max}}$, and $S_{\text{Ent}}$ represent the global average, global maximum feature, and information attention maps. $S_\alpha$, $S_\beta$, and $S_\gamma$ are initialised as 0 and learnable through training. $\text{Conv}^{7\times 7}$ is a $7 \times 7$ convolution kernel and $\sigma(.)$ is a sigmoid function. Moreover, the same as what we do in channel attention module, we normalise $S_{\text{Ent}}$ to $[-1, 1]$.

In our framework, we carry out a negative operation for $S_{\text{Avg}}$ and apply an element-wise addition for feature fusion. As shown in Figure 3a, the contributions of $S_{\text{Avg}}$ are different from that of $S_{\text{Max}}$ and $S_{\text{Ent}}$. In $S_{\text{Avg}}$, the response intensity of the background is significantly stronger than that of the object, while the opposite phenomenon appears for $S_{\text{Max}}$ and $S_{\text{Ent}}$. Conducting the negative operation on $S_{\text{Avg}}$ ($-S_{\text{Avg}}$) ensures that responses of different attention feature maps have the same characteristics. Using a 2-D Gaussian low-pass filter ($k \times k$) before performing the GMP is also useful to alleviate the adverse effects caused by noises to obtain $S_{\text{Max}}$. We set $k$ to be 5 as well.

We have verified two methods for fusing these attention maps. As shown in Figure 3c, element-wise addition is more advantageous as it brings a dynamic fusing method to adapt contributions of different attention maps. Moreover, we employ a $7 \times 7$ convolution to capture spatial neighbourhood features. In the end, we employ a sigmoid function $\sigma(.)$ on the spatial attention map. The pseudo-code of CAT is shown in Algorithm 1.

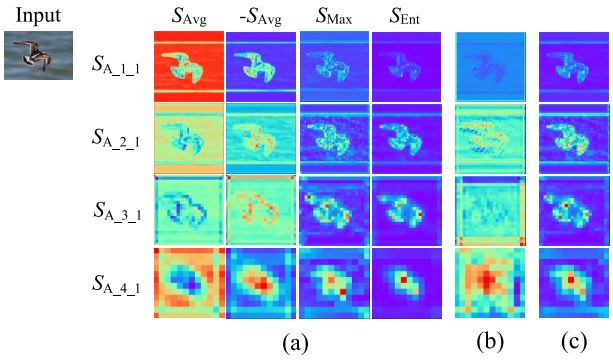



**FIGURE 3** Visualisation of spatial attention maps with different fusion methods. (a) Raw attention maps in the first block of each stage in ResNet50. Each column corresponds to the attention maps of global average pooling (GAP) ($S_{\text{Avg}}$, $-S_{\text{Avg}}$), GMP ($S_{\text{Max}}$), and global entropy pooling (GEP) ($S_{\text{Ent}}$). (b) Concatenated fusion attention maps and (c) Fusion attention maps from weighted summation. Best view in colour and zoom in.

**Algorithm 1 CAT, PyTorch-like**

```
# feat: Input feature maps, BxCxWxH
# w1, w2, w3: Trainable coefficients for three types
    of channel information.
# w4, w5, w6: Trainable coefficients for three types
    of spatial information.
# w7, w8: Trainable coefficients between channel and
    spatial scores.

att = CAT(feat)
feat = att * feat

def CAT(feat): # Compute CAT attention
   # Aggregate channel information via three ways.
   channel_avg = pool(feat, dim=(2, 3), method="Avg",
        keepdim=True) # BxCx1x1
   channel_max = pool(feat, dim=(2, 3), method="Max",
        keepdim=True) # BxCx1x1
   channel_entropy = pool(feat, dim=(2, 3), method="
        Entropy", keepdim=True)# BxCx1x1

   # Linear combination of channel information.
   channel_score = w1* channel_avg + w2 * channel_max+
        w3 * channel_entropy
   channel_score = Sigmoid(channel_score)

   # Aggregate spatial information via three ways.
   spatial_avg = pool(feat, dim=(1), method="Avg",
        keepdim=True) # Bx1xWxH
   spatial_max = pool(feat, dim=(1), method="Max",
        keepdim=True) # Bx1xWxH
   spatial_entropy = pool(feat, dim=(1), method="
        Entropy", keepdim=True) # Bx1xWxH

   # Linear combination of channel information.
   spatial_score = -w4 * spatial_avg + w5 *
        spatial_max+ w6 * spatial_entropy

   spatial_score = conv(spatial_score)
   spatial_score = Sigmoid(spatial_score)

   return w7 * channel_score + w8 * spatial_score
```

As shown in Figure 4, it is obvious that $-S_{\text{Avg}}$ pays more attention to the overall information of the input and considers the background and foreground equivalence at the same time. If the response intensity of a small object is high and the features obtained after the GAP are very small, the features of the small object may not be well preserved. $S_{\text{max}}$ extracts the global maximum feature of the input. It can retain the features of strong response such as edges or corners but it is susceptible to extreme values.

In $S_{\text{Ent}}$, although contour features of the object are not comprehensive as $S_{\text{max}}$, it has a stronger response to key texture features such as the tire of a vehicle, the beak of a bird, the wing of an aeroplane, and the bottle mouth etc. Consequently, the background of $S_{\text{Ent}}$ is cleaner than the other two attention maps and is less susceptible to background noises.

## 4 | EXPERIMENTS

In this section, we evaluate the proposed attention module CAT on MS COCO and Pascal VOC datasets for object detection with Faster-RCNN [44] and RetinaNet [45]

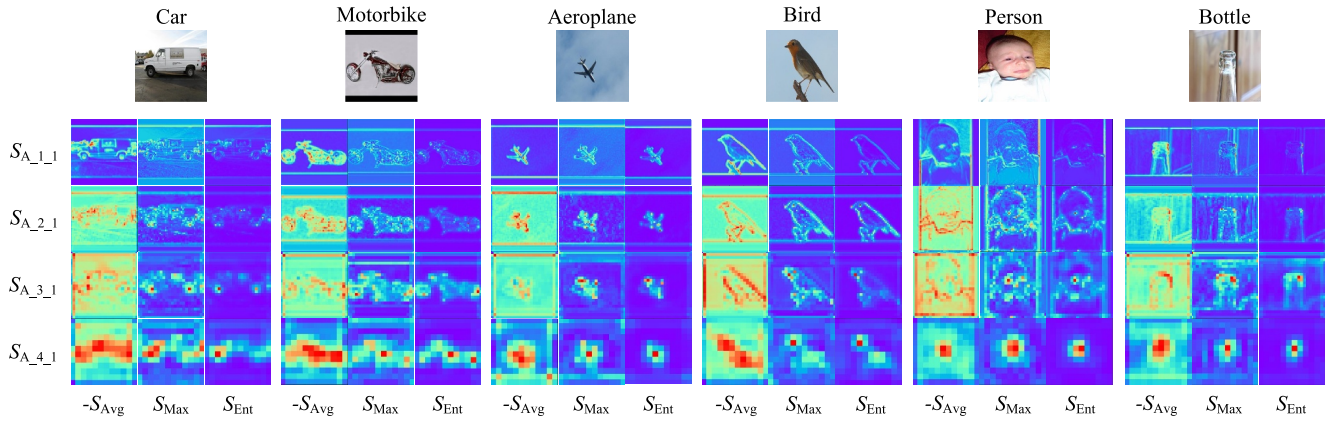



**FIGURE 4** The visualisation of spatial attention maps. From top to bottom, these are the spatial attention maps for the first block in Resnet50 four stages. Each column of each subgraph corresponds to the attention maps of global average pooling (GAP) ($-S_{Avg}$), GMP ($S_{Max}$), and global entropy pooling (GEP) ($S_{Ent}$). It is obvious that the background of the $S_{Ent}$ is cleaner than the $-S_{Avg}$ and $S_{Max}$ and has stronger response to the key texture features such as the tire of a vehicle, the beak of a bird, the wing of an aeroplane, and the bottle mouth. Best view in colour and zoom in.

frameworks. We also evaluate CAT for instance segmentation with Mask-RCNN [5], which is trained on MS COCO, with Cifar-100 dataset for image classification based on four backbones (ResNet50 [46], ResNet101 [46], MobileNetV2 [47], and ShuffleNetV2 [48]). Specifically, we conduct CAT module on ImageNet for image classification.

## 4.1 | Object detection experiments

We conduct object detection on Pascal VOC and MS COCO 2017 datasets and deploy all experiments on the Detectron2 platform. In this section, we adopt the average AP, $AP_{50}$, and $AP_{75}$ to evaluate the effects of the attention frameworks and use the Faster-RCNN and RetinaNet as detection architectures.

### 4.1.1 | Object detection on pascal VOC

Pascal VOC dataset contains 20 categories. The training set includes 22,136 images for VOC-2007 and VOC-2012 trainval and the test set includes 4952 images for VOC-2007. We adopt FasterRCNN as the detection architecture and load the model pretrained on ImageNet as our backbone network. In detail, we train the model on 4 Geforce GTX TITAN X GPUs and the batch size is set to 8. We take synchronous stochastic gradient descent (SGD) as an optimiser with a momentum of 0.9, weight decay of 0.0001, and total iteration number of 18,000. The initial learning rate is set to 0.02 and it drops by a factor of 10 after 12,000 and 16,000 iterations.

The experimental results are summarised in Table 1. The bold and underlined data indicate the optimal and sub-optimal results. It shows when applying Faster-RCNN as the basic detector, CAT is superior to the original ResNet by 2.07% and 1.23% in terms of AP when the network depth is 50 and 101. Meanwhile, CAT achieves 2.26% and 1.07% gains over SENet using ResNet50 and ResNet101. On the other hand, CAT outperforms the original ResNet by 1.04% in terms of AP when the network depth is 50 and 101 using the RetinaNet, and it also achieves better performance than all previous arts in terms of all three measures. Specifically, CAT improves CBAM over 0.72% and 1.57% for ResNet50 and ResNet101. When IoU = 0.75, CAT has a remarkable improvement in AP compared with other IoUs.

**TABLE 1** Object detection results of different attention methods on Pascal VOC test 2007

| Detectors | Methods | AP | $AP_{50}$ | $AP_{75}$ |
|---|---|---|---|---|
| Faster-RCNN | ResNet50 | 50.47 | 79.25 | 54.14 |
| | + SENet | 50.28 | 79.22 | 54.22 |
| | + CBAM | 50.72 | 79.52 | 54.32 |
| | + ECANet | 50.58 | 79.83 | 54.26 |
| | + CAT (Ours) | **52.54** | **80.96** | **57.46** |
| | ResNet101 | 53.10 | 80.98 | 58.45 |
| | + SENet | 53.26 | 81.00 | 58.96 |
| | + CBAM | 53.19 | 81.03 | 58.69 |
| | + ECANet | 53.16 | 81.28 | 58.06 |
| | + CAT (Ours) | **54.33** | **81.94** | **60.11** |
| RetinaNet | ResNet50 | 53.11 | 77.92 | 57.03 |
| | + SENet | 53.10 | 78.54 | 57.00 |
| | + CBAM | 53.43 | 77.17 | 57.81 |
| | + ECANet | 53.30 | 79.01 | 57.44 |
| | + CAT (Ours) | **54.15** | **79.93** | **58.66** |
| | ResNet101 | 54.38 | 78.79 | 58.68 |
| | + SENet | 54.94 | 79.57 | 59.36 |
| | + CBAM | 53.85 | 78.53 | 58.44 |
| | + ECANet | 55.12 | 79.70 | 59.57 |
| | + CAT (Ours) | **55.42** | **79.94** | **59.91** |

*Note*: Bolded values show the best performance of every matrices of different models after comparison in the tables.

### 4.1.2 | Object detection on MS COCO 2017

MS COCO 2017 includes 118,287 training images and 5000 validation images with 80 categories. Table 2 shows the performance of attention modules on MS COCO 2017. $AP_S$, $AP_M$, and $AP_L$ represent the AP value detected by small targets (area <32), medium targets (32 < area <96), and large targets (area >96). The number of training iterations is set to 90,000. The learning rate drops by a factor of 10 after 70,000 and 80,000 iterations. When applying RetinaNet as the basic detector, CAT outperforms the original ResNet by 1.76% and 0.97% in terms of AP when the network depth is 50 and 101. Meanwhile, CAT is superior to SENet and ECANet in terms of all precision measures. Besides, CAT outperforms the original ResNet by 1.72% and 1.14% in terms of AP when the network depth is 50 and 101 when using Faster-RCNN. Similarly, we can notice that when CAT detects large objects and IoU = 0.75, the corresponding AP obtained is the highest improvement compared to the baseline. This indicates that our method is more appropriate for scenarios that require rigorous object location and size, such as autonomous driving.

**TABLE 2** Object detection results of different attention methods on MS COCO val2017

| Detectors | Methods | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| Faster-RCNN | ResNet50 | 33.27 | 53.58 | 35.26 | 18.04 | 35.71 | 42.56 |
| | + SENet | 33.56 | 53.67 | 36.02 | 18.52 | 36.23 | 42.90 |
| | + CBAM | 33.37 | 53.67 | 35.61 | 19.38 | 35.84 | 43.60 |
| | + ECANet | 34.41 | 54.91 | 37.13 | 20.34 | 37.08 | 43.72 |
| | + CAT (Ours) | **34.99** | **55.80** | **37.69** | **20.55** | **37.86** | **44.86** |
| | ResNet101 | 35.64 | 55.71 | 38.49 | 20.55 | 38.70 | 45.37 |
| | + SENet | 35.70 | 55.97 | 38.54 | 20.54 | 38.99 | 45.61 |
| | + CBAM | **36.85** | **57.78** | 39.69 | **21.98** | **40.02** | 46.85 |
| | + ECANet | 36.18 | 56.64 | 38.85 | 21.50 | 39.16 | 46.44 |
| | + CAT (Ours) | 36.78 | 57.68 | **39.86** | 21.71 | 39.97 | **46.99** |
| RetinaNet | ResNet50 | 34.39 | 53.01 | 37.04 | 18.30 | 38.05 | 44.14 |
| | + SENet | 34.50 | 53.07 | 37.01 | 18.90 | 38.08 | 44.29 |
| | + CBAM | 34.38 | 52.95 | 36.60 | 19.60 | 38.21 | 44.22 |
| | + ECANet | 35.23 | 54.13 | 37.70 | 19.78 | 39.27 | 45.29 |
| | + CAT (Ours) | **36.15** | **55.38** | **38.53** | **21.40** | **40.12** | **47.10** |
| | ResNet101 | 35.48 | 53.70 | 37.89 | 20.07 | 39.02 | 45.04 |
| | + SENet | 35.49 | 54.04 | 38.09 | 20.67 | 39.06 | 45.45 |
| | + CBAM | 35.79 | 54.50 | 38.09 | 20.97 | 39.57 | 46.32 |
| | + ECANet | 36.05 | 54.92 | 38.47 | **21.17** | 40.08 | 45.67 |
| | + CAT (Ours) | **36.45** | **55.31** | **39.07** | **21.17** | **40.24** | **46.63** |

*Note*: Bolded values show the best performance of every matrices of different models after comparison in the tables.

## 4.2 | Image classification experiments

### 4.2.1 | Image classification on Cifar-100

To verify the classification effectiveness of CAT, we employ ResNet50, ResNet101, MobileNetV2, and ShuffleNetV2 as the base networks. Cifar-100 contains 100 categories and each category consists of 600 images (500 of the train set and 100 of the test set). We compare our module with SENet [12], CBAM [27], and ECANet [14] that are trained on the Cifar-100 training set and measure the Top-1 and Top-5 error on the test set. All programs are trained within 200 epochs on 1 Geforce GTX TITAN X GPU, and the batch size is 128. We use SGD with a momentum of 0.9 and a weight decay of 0.0005. The learning rate is initialised to 0.001 and drops by a factor of 10 every 50 epochs. The results are shown in Table 3. The CAT based on ResNet101 achieves 1.62% gain of Top-1 and 1.84% of Top-5. Besides, CAT improves CBAM on the accuracy of Top-1 and Top-5 over 0.31%, 0.74% and 0.35%, 0.44% for Mobilenetv2 and Shufflenetv2, respectively. And our CAT favourably improves all the strong baselines with negligible additional parameters. Such improvements demonstrate the efficiency of the CAT.

**TABLE 3** Image classification results of different attention methods on Cifar-100

| Methods | Top-1 | Top-5 | GMac | Params(M) |
|---|---|---|---|---|
| ResNet50 | 35.60 | 11.98 | 63.99 | 24.37 |
| + SENet | 34.93 | 12.17 | 64.08 | 26.22 |
| + CBAM | 36.28 | 12.78 | 64.17 | 26.22 |
| + ECANet | **34.36** | 11.86 | 64.08 | 23.71 |
| + CAT (Ours) | 34.78 | **11.35** | 64.20 | 26.22 |
| ResNet101 | 36.61 | 13.78 | 123.57 | 42.49 |
| + SENet | 36.24 | 13.16 | 123.72 | 47.44 |
| + CBAM | 37.35 | 13.87 | 123.90 | 47.44 |
| + ECANet | 36.76 | 13.55 | 123.91 | 47.45 |
| + CAT (Ours) | **34.99** | **11.94** | 123.91 | 47.45 |
| Mobilenetv2 | 48.24 | 19.33 | 2.42 | 2.37 |
| + SENet | 48.24 | 19.29 | 2.43 | 2.40 |
| + CBAM | 47.65 | 19.37 | 2.44 | 2.40 |
| + ECANet | 48.11 | 19.07 | 2.43 | 2.37 |
| + CAT (Ours) | **47.34** | **18.63** | 2.44 | 2.40 |
| Shufflenetv2 | 48.87 | 20.07 | 2.23 | 1.40 |
| + SENet | 48.09 | 19.09 | 2.24 | 1.40 |
| + CBAM | 48.41 | 19.08 | 2.25 | 1.40 |
| + ECANet | 48.07 | 19.62 | 2.24 | 1.36 |
| + CAT (Ours) | **48.06** | **18.64** | 2.25 | 1.40 |

*Note*: Bolded values show the best performance of every matrices of different models after comparison in the tables.

**TABLE 4** Accuracy comparisons with different attention methods on ImageNet

| Methods | Params(M) | FLOPs(G) | Top-1 | Top-5 |
|---|---|---|---|---|
| ResNet50 | 24.37 | 3.86 | 75.44 | 92.50 |
| + SENet | 26.77 | 3.87 | 76.86 | 93.30 |
| + CBAM | 26.77 | 3.87 | 77.34 | 93.69 |
| + ECANet | 24.37 | 3.86 | 77.48 | 93.68 |
| + CAT (ours) | 26.51 | 3.95 | **77.99** | **94.14** |
| ResNet101 | 42.49 | 7.34 | 76.62 | 93.12 |
| + SENet | 47.01 | 7.35 | 77.65 | 93.81 |
| + CBAM | 47.01 | 7.35 | 78.49 | 94.31 |
| + ECANet | 42.49 | 7.35 | 78.65 | **94.34** |
| + CAT (ours) | 44.15 | 7.81 | **78.74** | 94.32 |

*Note*: Bolded values show the best performance of every matrices of different models after comparison in the tables.

**TABLE 5** Instance segmentation results of different attention methods on MS COCO val2017 using Mask R-CNN

| Methods | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|
| ResNet50 | 33.45 | 53.54 | 35.93 | 15.98 | 35.58 | 48.18 |
| + SENet | 33.56 | 53.94 | 35.85 | 16.03 | 35.60 | 48.55 |
| + CBAM | 33.57 | 54.04 | 35.91 | 16.49 | 36.01 | 47.85 |
| + ECANet | 33.87 | 54.30 | 35.99 | 16.01 | 35.54 | 48.24 |
| + CAT (Ours) | **34.22** | **55.06** | **36.53** | **16.91** | **36.60** | **48.62** |
| ResNet101 | 35.06 | 55.38 | 37.55 | 16.85 | 37.36 | 50.74 |
| + SENet | 35.13 | 55.55 | 37.77 | 16.62 | 37.52 | 50.97 |
| + CBAM | 34.74 | 55.40 | 37.15 | 16.93 | 37.37 | 50.12 |
| + ECANet | 35.67 | 56.77 | 38.24 | **17.92** | **38.65** | 51.29 |
| + CAT (Ours) | **35.86** | **57.02** | **38.45** | 17.55 | 38.43 | **51.59** |

*Note*: Bolded values show the best performance of every matrices of different models after comparison in the tables.

### 4.2.2 | Image classification on ImageNet

To evaluate our CAT module on ImageNet classification, we employ ResNet50 and ResNet101 as backbone models. We optimise network parameters by SGD with a weight decay of 1e-4, momentum of 0.9, and mini-batch size of 256. We train all models for 100 epochs by setting the initial learning rate to be 0.1, which is decreased by a factor of 10 per 30 epochs. We compare our CAT module with SENet, CBAM, and ECANet on different backbones. As presented in Table 4, backbone networks with our CAT outperform all baselines significantly, demonstrating that CAT generalises well on various models in large-scale datasets. Moreover, CAT achieves the best Top-1 and Top-5 accuracy compared to other modules when the network depth is 50. Our CAT is competitive to other modules with negligible additional parameters used when the network depth is 101. Based on above results, CAT not only boosts the accuracy of baselines significantly but also favourably improves the performance of other modules. CAT also has the potential to boost its performance through further investigating the GEP operator and information fusing technique.

## 4.3 | Instance segmentation experiments

We deploy instance segmentation experiments on MS COCO 2017 with Mask-RCNN as the basic detector, and ResNet50 and ResNet101 as backbone models. The implementation details are generally the same as those in object detection, except that we employ 2 Geforce GTX TITAN X GPUs to train the model. As shown in Table 5, CAT outperforms the original ResNet by 0.77% and 0.8% in terms of AP when the network depth is 50 and 101. We are also aware that CAT has a better response to large objects. For ResNet50 as backbone, our model achieves the best performance in all indicators of segmentation tasks. For ResNet101 as backbone, which is accepted for the accuracy of small and medium targets, our model gets better AP than ECANet. These results prove the effectiveness of our model in the segmentation tasks.

**TABLE 6** The comparison of different attention methods with global entropy pooling (GEP) and without GEP on Pascal VOC validation 2012. CAT without exterior colla-factors and interior colla-factors, respectively, refer to the weights applied in the exterior collaboration approach for different attention modules and interior collaboration approach for three attention operators

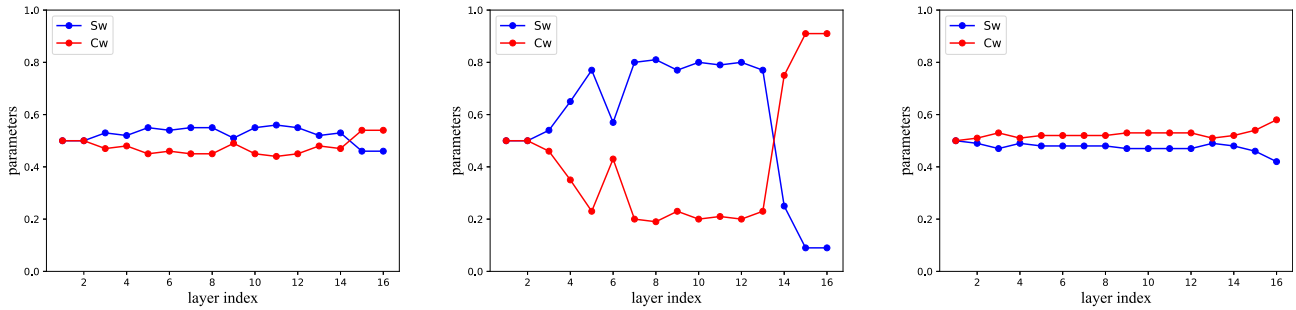
| Methods | AP | $AP_{50}$ | $AP_{75}$ |
|---|---|---|---|
| ResNet50 (baseline) | 40.06 | 71.13 | 41.61 |
| + Spatial | 40.21 | 71.09 | 41.20 |
| + Spatial w/o GEP | 40.16 | 71.01 | 41.17 |
| + Channel | 40.44 | 71.53 | 41.15 |
| + Channel w/o GEP | 40.38 | 71.31 | 41.1 |
| + Channel + spatial | 36.85 | 67.44 | 36.46 |
| + Channel + spatial w/o GEP | 36.61 | 67.05 | 36.29 |
| + Spatial + channel | 37.37 | 68.31 | 37.16 |
| + Spatial + channel w/o GEP | 37.12 | 67.92 | 36.81 |
| CAT w/exterior colla-factors | 41.81 | 72.99 | 43.53 |
| CAT w/exterior & interior colla-factors | **42.61** | **73.71** | **44.87** |

*Note*: Bolded values show the best performance of every matrices of different models after comparison in the tables.

## 4.4 | Ablation study

The ablation study on Pascal-VOC 2012 for object detection with Faster-RCNN verifies the effectiveness of the channel and spatial collaboration relationship proposed in this study as shown in Table 6.

We firstly introduce a GAP to extract the attention information and compare it with methods that use single




(a) Object Detection (b) Instance Segmentation (c) Image Classification

**FIGURE 5** Curves of weight parameters ($S_W$ and $C_W$) in different tasks with the change of embedding layers. Obviously, the collaborative relationships of these two parameters are different in diverse embedding hierarchies and tasks.

attention and sequentially arranged attention modules. We realise that the sequential method results are in degraded performance due to the uncontrolled interference between the channel and spatial information. The simple combination may not have an ideal performance because the result can inhibit the effect on some characteristics and can lead to counter-productive performance since channel and spatial active features perform in different ways. In addition, to verify the effectiveness of GEP on model performance improvement, we conducted experiments with and without GEP for different combinations of attention. Our method based on exterior collaboration tackles this problem by dynamically combining attention modules and obtains 41.81% in terms of AP. Moreover, we introduce an additional interior collaboration approach on the previous model for simultaneously combining GEP, GAP, and GMP operators. Its performance in AP rises to 42.61%.

## 4.5 | Visualisation

Line plots in Figure 5, according to priority, illustrate the change of parameters against embedding layers for object detection, instance segmentation, and image classification tasks with ResNet50 as backbone networks. The first two plots show similar patterns in low-level layers of the network.

$S_w$ values are usually larger than $C_w$ values due to the fact that low-level feature maps mainly extract various spatial features (such as textures, contours, and edges) to effectively guide the network to learn 'where' to focus. On the other hand, at high-level layers, $C_w$ values gradually exceed $S_w$ values because rich semantic information in high-level feature maps can effectively promote the network's learning of 'what' the object is.

However, for image classification task, $C_w$ is always higher than $S_w$ at all layer levels.

Such phenomenon reveals that classification networks pay more attention to objects' categories other than their locations. In conclusion, dynamically learning the relationship between channel and spatial modules for information fusion can improve the guidance ability of the overall attention framework.

# 5 | CONCLUSION

In this study, we propose CAT, a novel attention framework and design it for adaptively learning to collaborate the contributions of attention modules and pooling operators, respectively. Specifically, we introduce GEP to complement GAP and GMP for information extraction and study their traits' inherent collaboration relationship for better information fusion. Our CAT is a plug-and-play framework such that no expensive extra cost is required for applying it in networks. Extensive experiments demonstrate that CAT can improve networks' adaptability in various image hierarchies and tasks (such as classification, object detection, and instance segmentation) compared to previous state-of-the-art attention-based networks. We integrate our experiments and conclude that our CAT framework indicates the potential of elastically designed attention mechanisms in CNNs for computer vision tasks. We will develop our framework in more architectures to enhance its ability to explore interactive collaborations of attention modules.

## AUTHOR CONTRIBUTIONS

**Zizhang Wu**: Conceptualisation; Formal analysis; Methodology; Project administration; Software; Supervision; Writing—original draft; Writing—review & editing. **Man Wang**: Conceptualisation; Data curation; Formal analysis; Investigation; Methodology; Resources; Software; Validation; Visualisation; Writing—original draft; Writing—review & editing. **Weiwei Sun**: Conceptualisation; Data curation; Formal analysis; Investigation; Visualisation; Writing—original draft; Writing—review & editing. **Yuchen Li**: Data curation; Formal analysis; Investigation; Supervision; Validation; Visualisation; Writing—review & editing. **Tianhao Xu**: Resources; Software; Supervision; Validation; Visualisation; Writing—review & editing. **Fan Wang**. Resources; Software; Validation; Visualisation; Writing—review & editing. **Keke Huang**: Resources; Supervision; Validation; Visualisation; Writing—review & editing.

## ACKNOWLEDGEMENT

Open Access funding enabled and organized by Projekt DEAL.

## CONFLICT OF INTEREST

The author declares that there is no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## ORCID

*Tianhao Xu* https://orcid.org/0000-0002-7483-3940

## REFERENCES


1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
2. Zhou, B., et al.: Bbn: bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR (2020)
3. Song, G., Liu, Y., Wang, X.: Revisiting the sibling head in object detector. In: CVPR (2020)
4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
5. He, K., et al.: Mask R-CNN. In: ICCV (2017)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Bartlett, P.L., et al. (eds.) Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a Meeting Held December 3-6, 2012, pp. 1106–1114 (2012). https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
8. He, K., et al.: Deep residual learning for image recognition. In: CVPR (2016)
9. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: the missing ingredient for fast stylization. CoRR abs/160708022 (2016)
10. Sun, W., et al.: Acnet: attentive context normalization for robust permutation-equivariant learning. In: CVPR (2019)
11. Hou, Q., et al.: Strip pooling: rethinking spatial pooling for scene parsing. In: CVPR (2020)
12. Hu, J., et al.: Squeeze-and-excitation networks. In: CVPR (2017)
13. Wang, X., et al.: Non-local neural networks. In: CVPR (2018)
14. Wang, Q., et al.: Eca-net: efficient channel attention for deep convolutional neural networks. In: CVPR (2020)
15. Hu, J., et al.: Gather-excite: exploiting feature context in convolutional neural networks. In: NIPS (2018)
16. Li, H.: Channel locality block: a variant of squeeze-and-excitation. arXiv preprint arXiv: 190101493 (2019)
17. Xie, J., et al.: Channel attention with embedding Gaussian process: a probabilistic methodology. arXiv preprint arXiv: 200304575 (2020)
18. Zhang, Y., Min Fang, N.W., Wang, N.: Channel-spatial attention network for fewshot classification. PLoS One 14(12), e0225426 (2019). https://doi.org/10.1371/journal.pone.0225426
19. Jaderberg, M., et al.: Spatial transformer networks. In: NIPS (2015)
20. Khandelwal, S., Sigal, L.: Attentionrnn: a structured spatial attention mechanism. In: IEEE (2019)
21. Hu, X., et al.: Spatial pyramid attention network forimage manipulation localization. In: ECCV (2020)
22. Chen, L., et al.: Sca-cnn: spatial and channel-wise attention in convolutional networks for image captioning. In: CVPR (2017)
23. Gao, J., Wang, Q., Yuan, Y.'S.: Spatial-/channel-wise attention regression networks for crowd counting. Neurocomputing 363, 1–8 (2019). https://doi.org/10.1016/j.neucom.2019.08.018
24. Woo, S., et al.: Cbam: convolutional block attention module. In: ECCV (2018)
25. Fu, J., et al.: Dual attention network for scene segmentation. In: CVPR (2018)
26. Cao, Y., et al.: Gcnet: non-local networks meet squeeze-excitation networks and beyond. In: ICCV Workshops (2019)
27. Benassou, S.N., Shi, W., Jiang, F.: Entropy guided adversarial model for weakly supervised object localization. arXiv preprint arXiv: 200801786 (2020)
28. Li, X., Hu, X., Yang, J.: Spatial group-wise enhance: improving semantic feature learning in convolutional networks. arXiv preprint arXiv: 190509646 (2019)
29. Li, X., et al.: Selective kernel networks. In: IEEE (2019)
30. Wang, H., et al.: Parameter-free spatial attention network for person re-identification. arXiv preprint arXiv: 181112150 (2018)
31. Park, J., et al.: BAM: bottleneck attention module. In: British Machine Vision Conference 2018, BMVC 2018, pp. 147. BMVA Press (2018)
32. Qin, Z., et al.: Fcanet: frequency channel attention networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 783–792 (2021)
33. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, June 19–25, 2021, pp. 13713–13722. Computer Vision Foundation/IEEE (2021)
34. Zamir, S.W., et al.: Cycleisp: real image restoration via improved data synthesis. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, pp. 2693–2702. Computer Vision Foundation/IEEE (2020)
35. Zamir, S.W., et al.: Learning enriched features for real image restoration and enhancement. In: Vedaldi, A., et al. (eds.) Computer Vision – ECCV 2020 – 16th European Conference, Glasgow, UK, August 23-28, 2020. Proceedings, Part XXV. Vol. 12370 of Lecture Notes in Computer Science, pp. 492–511. Springer (2020)
36. Luo, J.H., Wu, J.: An entropy-based pruning method for cnn compression. arXiv preprint arXiv: 170605791 (2017)
37. Li, Y., et al.: Exploiting kernel sparsity and entropy for interpretable cnn compression. In: CVPR (2018)
38. Saito, K., et al.: Semi-supervised domain adaptation via minimax entropy. In: IEEE (2019)
39. Li, H., et al.: Pruning filters for efficient convents. CoRR, abs/160808710 (2016)
40. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: NIPS (2004)
41. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. arXiv preprint arXiv: 151106390 (2015)
42. Vu, T.H., et al.: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
43. Wan, W., et al.: Information entropy based feature pooling for convolutional neural networks. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, pp. 3404–3413. IEEE (2019)
44. Ren, S., et al.: Faster r-cnn: towards real-time object detection with region proposal networks. In: NIPS (2015)
45. Lin, T., et al.: Focal loss for dense object detection. In: ICCV (2017)
46. He, K., et al.: Deep residual learning for image recognition. CVPR (2016)
47. Sandler, M., et al.: Mobilenetv2: inverted residuals and linear bottlenecks. In: CVPR (2018)
48. Ma, N., et al.: Shufflenet v2: practical guidelines for efficient cnn architecture design. In: ECCV (2018)


**How to cite this article:** Wu, Z., et al.: CAT: Learning to collaborate channel and spatial attention from multi-information fusion. IET Comput. Vis. 1–10 (2022). https://doi.org/10.1049/cvi2.12166