PAPER

Recursive Multi-Scale Channel-Spatial Attention for Fine-Grained Image Classification

Dichao LIU[†], **Yu WANG**[††a)], ***Nonmembers***, **Kenji MASE**[†], ***and*** **Jien KATO**[††], ***Members***

SUMMARY Fine-grained image classification is a difficult problem, and previous studies mainly overcome this problem by locating multiple discriminative regions in different scales and then aggregating complementary information explored from the located regions. However, locating discriminative regions introduces heavy overhead and is not suitable for real-world application. In this paper, we propose the recursive multi-scale channel-spatial attention module (RMCSAM) for addressing this problem. Following the experience of previous research on fine-grained image classification, RMCSAM explores multi-scale attentional information. However, the attentional information is explored by recursively refining the deep feature maps of a convolutional neural network (CNN) to better correspond to multi-scale channel-wise and spatial-wise attention, instead of localizing attention regions. In this way, RMCSAM provides a lightweight module that can be inserted into standard CNNs. Experimental results show that RMCSAM can improve the classification accuracy and attention capturing ability over baselines. Also, RMCSAM performs better than other state-of-the-art attention modules in fine-grained image classification, and is complementary to some state-of-the-art approaches for fine-grained image classification. Code is available at https://github.com/Dichao-Liu/Recursive-Multi-Scale-Channel-Spatial-Attention-Module.

key words: *attention module, gated convolution, attention mechanism, image classification, fine-grained image recognition*

1. Introduction

Image classification, which refers to the labeling of images into a fixed set of categories, is a core problem in computer vision. As a fundamental, meaningful, and challenging subfield of image classification, fine-grained image classification (FGIC) has attracted much attention in recent years. FGIC aims to distinguish images belonging to different subcategories within the same basic-level category, e.g., different species of birds or different models of cars. In the real world, FGIC is the fundamental technology for a broad range of applications, such as automatic biodiversity monitoring, road vehicle monitoring, and so on. However, FGIC is a very challenging task, and the challenges are principally related to two characteristics of its own: inter-class similarity and intra-class variance.

Many previous studies have shown that accurately identifying visual attention (i.e., discriminative visual information) is the key to mitigate the adverse effect caused by inter-class similarity and intra-class variance [1]–[16]. Some of those studies utilize extra manual bounding boxes or part annotations to localize attentional regions, which improves the classification accuracy but is labor-intensive and limits the practicality of real-world applications [12]–[16]. Some other studies localize attentional regions with weakly supervised localization schemes [1]–[7], [17], [18]. By doing so, those studies have achieved promising results while avoiding the human effort for labeling bounding boxes or part annotations.

However, the approaches using weakly supervised localization schemes are facing the problem of the high overhead of computation time, memory, etc. Prior studies predominately utilize convolutional neural networks (CNNs) for localizing and recognizing the attentional regions. Typically in previous work, a localization network is used to learn the regions of the object shared among the same categories, and a classification network is used to learn discriminative features from the localized objects [17]–[19]. Compared with classifying the raw input images with a single classification network (i.e., without attention locating), the introduction of the localization network brings performance improvement as well as much extra overhead. For example, if the localization and classification networks have the same backbone, the overhead is at least doubled while using one localization network together with one classification network [17].

Moreover, for FGIC tasks, a single-scale attentional region cannot cover all the discriminative visual information of each image (as shown in Fig. 1). Consequently, many approaches localize multi-scale attentional regions, which provide complementary visual information [1]–[11]. Such a strategy improves the classification performance but causes huge overhead, which is needed for localizing multiple regions and classifying the multiple localized regions. For example, Zhang *et al.* [1] firstly roughly localize an initial attentional region containing important objects by weakly supervised object detection and segmentation using Mask R-CNN [20] and CRF-based segmentation [21]. Then they estimate and search multiple attentional regions, which can be of various scales, to provide complementary information to the initial attentional region obtained in the former step. The aggregation of the features extracted from the multi-scale attentional regions is proved to have better classification performance than the features extracted with single-scale attention regions. However, while improving classification accu-

Manuscript received August 6, 2021.
Manuscript revised November 12, 2021.
Manuscript publicized December 22, 2021.

[†]The authors are with the Graduate School of Informatics, Nagoya University, Nagoya-shi, 464–8601 Japan.
[††]The authors are with the College of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525–8577 Japan.
a) E-mail: ywang@nagoya-u.jp (Corresponding author)

DOI: 10.1587/transinf.2021EDP7166

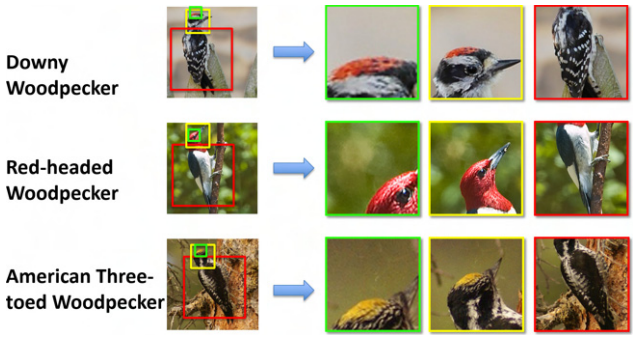


Fig. 1 Examples of multi-scale attentional regions for the images of different woodpeckers. Different scales of attentional regions can capture different objects, such as nape, head, and body. All the information is important for distinguishing different woodpeckers. For example, Downy Woodpecker has a red nape. Red-headed Woodpecker has a bright-red head. American Three-toed Woodpecker has a black and white barred back and white breast. For capturing multi-scale attentional information, many previous fine-grained image classification approaches focus on additional mechanisms acting as the output component of the backbone CNNs to crop multiple attentional regions [1]–[11]. Then, the outputted attentional regions are categorized by other backbone CNNs specifically for classification use. Differently, our proposed module can be embedded inside the backbone CNNs, and it refines the deep feature maps by exploring and utilizing multi-scale attentional information.

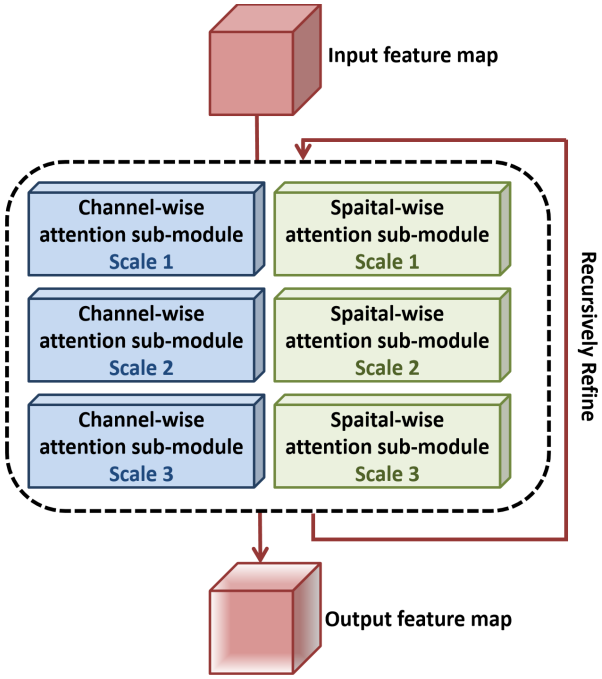


Fig. 2 Illustration of the main ideas of our work. The proposed attention module has six sub-modules: three-scale channel-wise sub-modules and three-scale spatial-wise sub-modules. The input feature map is recursively refined through the six sub-modules for a predetermined number of times to output the finally refined feature map.

racy, this approach requires multiple steps including roughly localizing an initial region, proposing multiple complementary regions, extracting features from attentional regions, and aggregating the features. Each step requires different network models such as Mask R-CNN [20], standard CNNs, LSTM [22], etc. Thus this approach is not only complicated but also requires a huge overhead.

To overcome the above-mentioned challenges, we propose a novel recursive multi-scale channel-spatial attention module (RMCSAM) for FGIC. Our approach follows the experience that multi-scale attention information is effective for FGIC tasks. However, note that RMCSAM exploits multi-scale attention information by the the fully-connected (FC) layers with multiple channel sizes and convolutional operations with multiple kernel sizes. This makes our approach different from the previous multi-scale attention learning strategies, which captures multi-scale attention information by cropping multi-size regions on the input image [5], [7], [8], [10], [11] or learning multi-size feature maps [3].

The proposed RMCSAM follows the success of the previous research on attention modules [23]–[27]. Attention modules refer to a set of insertable modules that enhance the feature representations generated by standard convolutional layers by giving weights among the channels or spatial locations of the feature. For example, the squeeze-and-excitation module (SE module) [24], which is one of the most prominent attention mechanisms, performs channel-wise attention by extracting global information from each channel and then generating a set of weights for each channel. By doing so, the SE module provides a boost of classification accuracy with a low additional overhead. The point-wise spatial attention module (PSA module) [23] is another typical example. The PSA module uses self-adaptively predicted attention maps to aggregate long-range contextual information within images, which boosts the performance for the scene parsing task. These attention modules are generally insertable into different network architectures and able to improve the networks' focus on important information.

The RMCSAM is designed as an attention module that explores multi-scale attention information and uses the explored information to enhance the deep features learned in the FGIC task. As an attention module, RMCSAM can be easily placed inside various backbone CNNs, such as ResNet [28] or VGG models [29]. Trained together with the backbone CNNs, RMCSAM improves the correspondence to attentional information for better classification accuracy. Clearly, our approach is different from previous FGIC approaches, which mainly design mechanisms placed as the output parts of the backbone CNNs yielding attentional information (e.g., attentional regions) [1]–[11].

Specifically, as shown in Fig. 2, the main ideas of the proposed RMCSAM are summarized as follows:

- Rather than localization and categorization of attentional regions, which is commonly used in previous FGIC approaches [1]–[16], we focus on developing an insertable attention module for the FGIC task.
- We design the proposed attention module to explore both channel-wise and spatial-wise attention. For the channel-wise attention, we firstly spatially pool the

given features and then use the pooled features to compute channel-wise weights with a set of fully connected (FC) layers. For the spatial-wise attention, we firstly pool the given features along the channel axis and then use the pooled features to compute spatial-wise weights with a set of convolutional layers. The features learned with the channel-wise and spatial-wise attention sub-module are aggregated by average.

- Following the prior experience that multi-scale attention is very important and effective for FGIC, we design the proposed attention module to perform three-scale channel-wise and spatial-wise attention. The different scales of the channel-wise sub-modules are defined with different numbers of the neurons in the FC layers within the sub-modules. The different scales of the spatial-wise sub-modules are defined with different kernel sizes in the convolutional layers within the sub-modules. The features refined by different scales of sub-modules are aggregated by average. Even though the proposed module is designed to perform three-scale channel-wise and spatial-wise attention, the whole module is still very lightweight because each sub-module only requires a small number of parameters.
- We design the proposed attention module to progressively refine the learned attention. Starting from the feature map outputted by a standard convolutional layer, we design a cyclically learning scheduler to generate more effective features by iteratively treating the output of the former learned attention module as the input of the current attention module. The attention modules in the different stages share the same parameters.

Our contributions can be summarized as follows:

- We propose a simple yet effective attention module that can explore multi-scale attention with negligible overhead for FGIC tasks.
- The proposed module can be easily inserted into standard CNNs and improve the classification accuracy for FGIC.
- We evaluate the proposed module on two benchmarks: CUB-200-2011 [30] and Stanford Cars [31]. We have validated the effectiveness of the design of the proposed attention module through extensive ablation studies. Experimental results show that RMCSAM can improve the classification accuracy and attention capturing ability over baselines. Also, RMCSAM outperforms other state-of-the-art attention modules [24]–[27] in FGIC tasks.
- As an insertable attention module, our approach can be combined with some previous approaches achieving state-of-the-art accuracy in the FGIC task [32], [33]. By combining our approach with the PMG framework [32], we achieve the best accuracy on the Stanford Cars and surpass the previous best accuracy obtained with the Resnet50 backbone on the CUB-200-2011.

2. Related Studies

2.1 Multi-Scale Attentional Region Learning for Fine-grained Image Classification

Effectively exploring attention is extremely crucial in FGIC tasks, and many previous studies propose to localize and classify multi-scale attentional regions that provide complementary and comprehensive information [1]–[11]. Early studies mainly rely on manual object bounding boxes or part annotations. For example, Xie *et al.* [11] propose to utilize the manual object bounding boxes to obtain image segmentation and give a descriptive image representation by building mid-level structures on the segmented regions. However, collecting manual annotations is time-consuming, labor-intensive and not feasible for real-world applications.

To the best of our knowledge, Xiao *et al.* [10] propose the first work using a multi-scale attention strategy for FGIC without using any manual object bounding boxes or part annotations (also mentioned in [5], [8]). In [10], the researchers propose a two-level attention model: object-level attention is to localize regions containing target objects for classification, and part-level attention is to localize small parts of the objects that are helpful for classification. The attention is localized by using a CNN to select patches relevant to the basic-level category, thus the dependence on manual object bounding boxes or part annotations is avoided.

Recently, the strategy of localizing and classifying multi-scale attentional regions plays a more and more crucial role in FGIC tasks. Fu *et al.* [7] recursively localize attentional regions from coarse to fine with the CNNs adapted for region proposal. The prediction of fine-scale attentional regions is given by taking the prediction of coarse-scale attentional regions as a reference.

Ding *et al.* [2] propose to localize pyramidal regions of interest (ROIs) in a weakly supervised manner by building a dual pathway hierarchy on the basic CNN following a bottom-up attention pathway and a top-down feature pathway. Then the localized regions are used to refine low-level features by erasing the most discriminative region to encourage the network to find more discriminative regions and generating major regions by merging all the ROIs.

Rao *et al.* [3] propose to localize multi-scale attentional regions and remove excessive unimportant regions by using the deep features learned with a Feature Pyramid Network (FPN). Zhang *et al.* [1] propose to estimate and search multiple attentional regions providing complementary information to an initial region localized by using Mask R-CNN and CRF-based segmentation. Then they use standard CNNs to extract features from those regions and lastly use an LSTM to aggregate the features.

The latest success of transformer in some other fields [34], [35] has influenced the attention-based research in the FGIC field. A transformer is a deep learning model giving attention weights to each element of the input data.

It was originally proposed for the natural language processing task [36] and has been adapted for computer vision tasks [37], [38]. He et al. [38] proposed a transformer-based multi-attention model specifically for FGIC use, which is called TransFG. TransFG first splits the input images into small regions, and the regions are projected into feature space by the transformer encoder. Thereafter, TransFG combines all raw attention weights of the transformer to be an attention map and uses the attention map as guidance for selecting discriminative regions. TransFG does not output the selected regions and then explore information from the selected regions. On the contrary, TransFG intuitively considers the attention link of the transformer as an indicator of attention. Specifically, before the last Transformer Layer, TransFG utilizes a part selection module (PSM) to select the tokens that correspond to the discriminative regions and only feed the selected tokens to the last transformer layer.

Though bringing a boost in terms of classification accuracy, these approaches have the problem of high overhead for memory, computation cost, etc., because the localization of multi-scale regions inevitably requires a high cost. TransFG does not require directly localizing the attention regions by outputting the regions and achieves state-of-the-art accuracy among the studies mentioned in this subsection. However, the backbone transformer, which itself has a extremely heavy computation overhead, together with the complicated part selection module, makes TransFG require much more parameters, GFLOPs, and time cost than the proposed approach. Different from these studies, our work provides an insertable, lightweight, and general module, which can be inserted into standard CNNs and only requires a little extra overhead.

2.2 Other Fine-Grained Image Classification Approaches

Besides exploring attentional regions, there are some state-of-the-art FGIC approaches focusing on other strategies.

Decision tree. Decision tree refers to a process that selects the appropriate directions based on the characteristic of features [39]. The inherent interpretability of decision tree has attracted much interest in adapting it for the FGIC task. Nauta et al. [40] proposed the Neural Prototype Tree (ProtoTree) that consists of a CNN backbone followed by a binary tree structure. ProtoTree can be trained end-to-end and locally explain each prediction by describing a decision path. Ji et al. [41] proposed to combine convolutional operations along edges of the tree structure and determines the decision path using the routing functions in each node. The convolutional operations generate the representations of objects, and the tree structure provides a feature learning process to exploit the representations.

Exploring the relation between deep feature elements. The intrinsic interrelationship between feature elements contains useful semantic information. Xu et al. [42] proposed a discrimination-aware mechanism (DAM) that improves the deep features conditioned to the analysis on the relation between deep feature elements. DAM can find the feature elements that are not well-learned and refine such elements for better FGIC performance. Zhao et al. [43] proposed a graph-based relation discovery (GaRD) approach to explore the high-order relationships among deep feature elements in the FGIC task. Given an input image, GaRD first generates a high-dimensional feature bank that is regularized with high-order constraints. Then GaRD utilizes a graph-based aggregating procedure to explore the relation between high-order elements of the feature bank and produce a low-dimensional feature representation.

Progressive learning. In the FGIC field, progressive learning approaches generally first divide a backbone CNN into several segments, and each segment progressively learns features and gives the prediction. Thereafter, the features learned by each segment are concatenated to give an overall prediction. Du et al. [32] proposed the Progressive Multi-Granularity (PMG), which uses a jigsaw puzzle generator to produce the images with different levels of granularity and then learns cross-granularity information by progressive learning. Zhang et al. [33] proposed to explore the similarity between the images of the same category and the difference between the images of different categories.

These approaches achieve state-of-the-art accuracy but suffer from huge computational expenses caused by their sophisticated architecture [40]–[43] or multi-stage framework [32], [33]. Moreover, as an insertable module, our approach is complementary to some state-of-the-art frameworks, such as [32], [33], and our approach can improve the accuracy of them.

2.3 Insertable Attention Modules

Attention modules are designed to make CNNs learn to focus on the important information and ignore unuseful information by imitating the human visual attention mechanism [24], [25], [27]. Humans tend to process an image by regarding it as a sequence of partial glimpses and selectively concentrate on informative parts, instead of processing a whole scene at once. Inspired by this fact, there have been emerging efforts to incorporate attention modules into CNNs for improving classification accuracy in large-scale classification tasks, such as ImageNet [44].

These attention modules generally consist of some pooling layers, 2D-convolutional layers, FC layers, and a sigmoid function at the end to generate a mask of the input feature map. For example, the SE module [24] squeezes global spatial information with 2D-pooling and excites the squeezed information into a set of channel weights to capture channel-wise dependencies. The success of the SE module is succeeded by many studies. CBAM [27] uses a similar idea to the SE module to capture channel-wise attention and introduces spatial-wise attention encoding implemented by 2D-convolutional layers with large-size kernels. Dai *et al.* [25] propose channel-wise attention in multiple scales by varying the spatial pooling size. The proposed module can be used to fuse deep features.

Different from the above-mentioned attention mod-

ules, our module can explore multi-scale attention of the input feature maps in both channel-wise and spatial wise. The multi-scale channel-wise attention in our work is implemented by using different numbers of the hidden units within the channel-wise sub-modules, which makes it different from the multi-scale channel-wise attention proposed in [25]. FC layers of different numbers of the hidden units can compress the features into different scales [46], the compressed features can then be used to generate multi-scale channel-wise dependencies. In this way, our work requires less overhead than [25] to explore multi-scale channel-wise attention. Moreover, our module can recurrently refine the features a predetermined number of times before outputting the final refined features.

3. Proposed Approach

In this section, we introduce the proposed RMCSAM in detail. As shown in Fig. 3, given an input feature map, RMCSAM first processes it via six sub-modules: three channel-wise sub-modules in different scales and three spatial-wise sub-modules in different scales. The processed feature maps are aggregated to be an output feature map. Thereafter, the output feature map is treated as the input feature map of the six sub-modules and processed again by the six sub-modules. This process is repeated a predetermined number of times to obtain the final refined feature map.

3.1 Multi-Scale Channel-Wise Attention Sub-Modules

The multi-scale channel-wise attention sub-modules are used to exploit inter-dependencies among the channels of a given feature map. In CNNs, each channel of a feature map acts as an object detector [47]. Consequently, channel-wise attention tells what objects are discriminative or unimportant for distinguishing a given image [27]. For example, bird head and bird claw are generally discriminative objects for distinguishing different bird species, and some other objects, such as tree branches, are not important for classification. We describe the detailed operation of the multi-scale channel-wise attention sub-modules below.

Firstly, consider a single-scale channel-wise attention sub-module, which is implemented similarly to the SE module [24]. Let $X \in \mathbb{R}^{H\times W\times C}$ be an input feature map generated by the former layer within a CNN. H, W and C respectively represents the spatial height, width and number of channels. Let $\Omega_r^{chl}(.)$ denote the function of the single-scale channel-wise attention sub-module. Note that r is a manual hyper parameter controlling the scale of the attention module, and it will be introduced in detail later in this subsection. An overview of the function of the single-scale channel-wise attention sub-module can be summarized as: output a 1D channel-wise weighted mask $M_r^{chl} \in \mathbb{R}^{1\times1\times C}$ and then put M_r^{chl} on X for emphasizing the discriminative channels and de-emphasizing the unimportant channels. A mathematical definition of $\Omega_r^{chl}(.)$ can be given as:

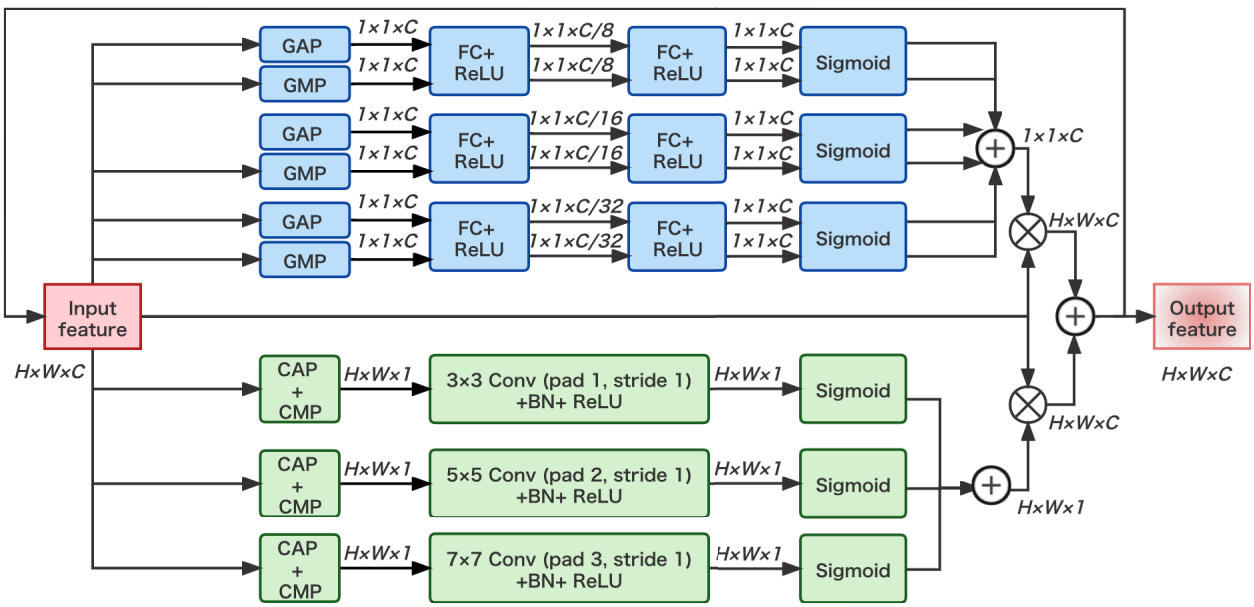


Fig. 3 Illustration of the proposed recursive multi-scale channel-spatial attention module (RMCSAM). "GAP" and "GMP" respectively represent the global average and max pooling. "CAP" and "CMP" respectively represent the channel-wise average and max pooling. "FC" and "Conv" respectively represent fully-connected layer and convolutional layer. "BN" and "ReLU" respectively represent batch normalization [45] layer and ReLU layer."⊕" represents element-wise sum. "⊗" denotes broadcast element-wise multiplication. The feature maps are denoted as feature dimensions, e.g. "$H\times W\times C$" denotes a feature map with height H, width W and channel number C.

$$X_r^{chl} = \Omega_r^{chl}(X) = X \otimes M_r^{chl}, \tag{1}$$

where X_r^{chl} denotes the refined feature map outputted by the single-scale channel-wise sub-attention module, and $\otimes$ denotes element-wise production. During $\otimes$, the values of M_r^{chl} are broadcasted along the spatial dimension to make M_r^{chl} have the same size as X.

M_r^{chl} is obtained from X with a set of pooling, fully connected (FC), and sigmoid operations. As average-pooled and max-pooled features provide complementary information [27], we first use both global average pooling and global max pooling to spatially shrink X to generate 1D channel-wise descriptors $D^{avg} \in \mathbb{R}^{1\times1\times C}$ and $D^{max} \in \mathbb{R}^{1\times1\times C}$ as:

$$d_c^{avg} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{i,j,c}, \tag{2}$$

$$d_c^{max} = \max_{i=1}^{H} \max_{j=1}^{W} x_{i,j,c}, \tag{3}$$

where d_c^{avg} and d_c^{max} are respectively the values in the channel c ($c \in \{1, 2, 3, \ldots, C\}$) of D^{avg} and D^{max}. $x_{i,j,c}$ denotes the value at the spatial location (i, j) in the channel c of X. Then both D^{avg} and D^{max} are processed by two successive FC layers as:

$$\begin{aligned} D^{avg\prime} &= \Phi^{avg}(D^{avg}) \\ &= f^{ReLU}(\phi_C^{avg}(f^{ReLU}(\phi_{\frac{C}{r}}^{avg}(D^{avg})))) \end{aligned} \tag{4}$$

$$\begin{aligned} D^{max\prime} &= \Phi^{max}(D^{max}) \\ &= f^{ReLU}(\phi_C^{max}(f^{ReLU}(\phi_{\frac{C}{r}}^{max}(D^{max})))) \end{aligned} \tag{5}$$

where $\Phi^{avg}(.)$ denotes the layers processing D^{avg}, and $\Phi^{max}(.)$ denotes the layers processing D^{max}. $f^{ReLU(.)}$ denotes ReLU operation. $\Phi^{avg}(.)$ and $\Phi^{max}(.)$ share the same parameters in order to reduce overhead. For both $\Phi^{avg}(.)$ and $\Phi^{max}(.)$, the output size of the first FC layer (i.e., $\phi_{\frac{C}{r}}^{avg}(.)$ or $\phi_{\frac{C}{r}}^{max}(.)$) is set as $1 \times 1 \times \frac{C}{r}$, and this FC layer is used to compress the channel-wise information of D^{avg} or D^{max} into a certain scale. The output size of the second FC layer (i.e., $\phi_C^{avg}(.)$ or $\phi_C^{max}(.)$) is set as C, and this FC layer makes the output descriptor have the same size of channels of X (so that the element-wise multiplication in Eq. (1) can be implemented).

Thereafter, M_r^{chl} is obtained as:

$$M_r^{chl} = \sigma(D^{avg\prime}) + \sigma(D^{max\prime}) \tag{6}$$

where σ represents the sigmoid operation, which makes each value range from 0 to 1 and thus gives the importance of each channel of X. The refined feature map X_r^{chl} can be obtained by substituting the M_r^{chl} obatined in Eq. (6) into Eq. (1).

The multi-scale channel-wise attention is obtained with different r. r controls the output size of $\phi_{\frac{C}{r}}^{avg}(.)$ and $\phi_{\frac{C}{r}}^{max}(.)$. A smaller output size makes the output information more compressed and gives a more abstract representation of the input descriptor (i.e., D^{avg} or D^{max}). A larger output size makes the output keep more information and gives a more detailed and inclusive representation of D^{avg} or D^{max}. In order to obtain all-sided channel-wise attention information, we build up multi-scale channel-wise attention sub-modules by using three different r: 8, 16 and 32. The refined feature map outputted by the multi-scale channel-wise attention sub-modules is defined as:

$$\begin{aligned} X_{multi}^{chl} &= \boldsymbol{\Omega^{chl}}(X) \\ &= \Omega_8^{chl}(X) + \Omega_{16}^{chl}(X) + \Omega_{32}^{chl}(X). \end{aligned} \tag{7}$$

3.2 Multi-Scale Spatial-Wise Attention Sub-Modules

The multi-scale spatial-wise sub-attention module are used to exploit inter-dependencies among the spatial locations of a given feature map. Spatial-wise attention tells the spatial location of the discriminative objects. As introduced in Sect. 3.1, channel-wise attention tells what objects are discriminative for classification. Thus, these two types of attention are complementary to each other. We describe the detailed operation of the multi-scale spatial-wise attention sub-modules below.

Firstly, consider a single-scale spatial-wise attention sub-module. Similar to the formulation in Sect. 3.1, $X \in \mathbb{R}^{H\times W\times C}$ denotes an input feature map, and $\Omega_k^{spat}(.)$ denote the function of the single-scale spatial-wise attention sub-module. Note that k is a manual parameter controlling the scale of the attention sub-module, and it will be introduced in detail later in this subsection. An overview of the function of the single-scale spatial-wise attention sub-module can be summarized as: output a 2D spatial-wise weighted mask $M_k^{spat} \in \mathbb{R}^{H\times W\times 1}$ and then put M_k^{spat} on X for emphasizing the discriminative spatial locations and de-emphasizing the unimportant spatial locations. A mathematical definition of $\Omega_k^{spat}(.)$ can be given as:

$$X_k^{spat} = \Omega_k^{spat}(X) = X \otimes M_k^{spat}, \tag{8}$$

where X_k^{spat} denotes the refined feature map outputted by the single-scale spatial-wise attention sub-module, and during $\otimes$, the values of M_k^{spat} are broadcasted along the channel dimension to make M_r^{spat} have the same size as X.

M_r^{spat} is obtained from X with a set of operations including channel-wise pooling, 2D convolution, and sigmoid. The first step for obtaining M_r^{spat} is to shrink X along the channel dimension to generate 2D spatial-wise score maps $S^{avg} \in \mathbb{R}^{H\times W\times 1}$ and $S^{max} \in \mathbb{R}^{H\times W\times 1}$ as:

$$s_{i,j}^{avg} = \frac{1}{C} \sum_{c=1}^{C} x_{i,j,c}, \tag{9}$$

$$s_{i,j}^{max} = \max_{c=1}^{C} x_{i,j,c}, \tag{10}$$

where $s_{i,j}^{avg}$ and $s_{i,j}^{max}$ are respectively the values at the location (i, j) of S^{avg} and S^{max}. $x_{i,j,c}$ denotes the value at the spatial location (i, j) in the channel c of X. Then S^{avg} and

S^{max} are processed as:

$$S^{'} = \psi_{k\times k\times 2\times 1}(f^{cat}(S^{avg}, S^{max})), \tag{11}$$

where f^{cat} denotes a channel-wise concatenation operation. $\psi_{k\times k\times 2\times 1}(.)$ denotes a 2D convolutional layer whose kernal size is $k\times k\times 2\times 1$, and this layer is used to encode the spatial-wise information of each $k \times k$-size region inside S^{avg} and S^{max}. The padding size of $\psi_{k\times k\times 2\times 1}(.)$ is set as $\frac{k-1}{2}$ and the stride is set as 1. Consequently, $\psi_{k\times k\times 2\times 1}(.)$ does not change the spatial size of the input feature map.

Thereafter, M_k^{spat} is obtained as:

$$M_k^{spat} = \sigma(f^{ReLU}(f^{BN}(S^{'})), \tag{12}$$

where $f^{BN}(.)$ denotes batch normalization operation [45]. The refined feature map X_k^{spat} can be obtained by substituting the M_k^{spat} obatined in Eq. (12) into Eq. (8).

The multi-scale spatial-wise attention is obtained with different k. k controls the kernel size of $\psi_{k\times k\times 2\times 1}(.)$. That is, k decides each value of M_k^{spat} to be corresponding to how large a region in S^{avg} and S^{max}. A 2D convolutional layer of a smaller kernel size has smaller receptive fields and thus can capture more local information and more detailed clues. A 2D convolutional layer of a bigger kernel size has bigger receptive fields and thus can "see" more information at once and capture relatively more global information, such as the dependencies among some local patterns. In order to obtain comprehensive spatial-wise attention information, we build up multi-scale spatial-wise attention sub-modules by using three different k: 3, 5 and 7. The refined feature map outputted by the multi-scale spatial-wise attention sub-modules is defined as:

$$\begin{aligned} X_{multi}^{spat} &= \boldsymbol{\Omega}^{\boldsymbol{spat}}(X) \\ &= \Omega_3^{spat}(X) + \Omega_5^{spat}(X) + \Omega_7^{spat}(X). \end{aligned} \tag{13}$$

3.3 Recursive Refinement

Our module recursively refines the given feature maps to focus on the discriminative visual information more finely. Let T denote how many times we refine the feature maps, and let X_t^{ref} ($t \in \{0, 1, 2, 3, \ldots, T\}$) denote the feature map outputted at time t. We recursively refine the feature map by treating the output at time $t-1$ as the input of time t. A mathematical definition is given as:

$$\begin{aligned} X_0^{ref} &= X, \\ X_t^{ref} &= \boldsymbol{\Omega}^{\boldsymbol{chl}}(X_{t-1}^{ref}) + \boldsymbol{\Omega}^{\boldsymbol{spat}}(X_{t-1}^{ref}). \end{aligned} \tag{14}$$

4. Experiments

4.1 Experimental Settings

To evaluate the effectiveness of our approach, we carried out experiments on two widely-used, competitive and standard benchmarks, namely CUB-200-2011 [30] and Stanford Cars [31]. CUB-200-2011 is a benchmark of bird images across 200 different species. There are totally 11,788 images, 5994 for training and 5794 for testing. Stanford Cars is a benchmark of car images across 196 car models. There are totally 16,185 images, 8,144 for training and 8,041 for testing.

As our approach is actually a lightweight insertable module, we compare the FGIC performance of the standard networks without the proposed module, with the proposed module, with other state-of-the-art attention modules. Besides, we also compare our approach with the latest state-of-the-art FGIC approaches [32], [33], [38], [40]–[43]. Following the experience in previous studies [25], [27], we insert the proposed module after the final convolutional block of each network. In order to perform apple-to-apple comparisons, we reproduced all the evaluated networks with the same training and testing configuration.

For the training procedure, we resize the images to make the shorter side be 512, while keeping the aspect ratio being unchanged. Then we randomly crop a 448×448 part augmented with random flipping as the input. Consequently, the GFLOPs in this paper are reported by computing with 448×448 input. For the testing procedure, we resize the images in the same way as the training procedure but use center cropping to obtain the 448×448 input images. For keeping the interference factors as few as possible and obtaining a stable result, we evaluate the time cost of the proposed approach as well as other approaches by handling a group of eight input images (unless otherwise specified), i.e., an $8 \times 3 \times 448 \times 448$ tensor, with a single Nvidia GTX 1080 Ti.

Regarding the parameter initialization, we use the network backbones pre-trained on the ImageNet [44] (provided by PyTorch [48]) and then fine-tune them on the fine-grained image classification datasets. The inserted RMCSAM, as well as other attention modules, are randomly initialized. However, in Sect. 4.6, to further improve the accuracy, we also implement the experiment of pre-training RMCSAM with the Resnet50 backbone on the ImageNet once before the fine-tuning (see more training details in Sect. 4.6). For all the other experiments,we use same experimental configuration:

- We reproduce all the experiments 10 times and report the average accuracy.
- We train all the networks using standard Stochastic Gradient Descent (SGD) with the momentum of 0.9, batch size of 32, weight decay of 5×10^{-4}, learning rate of 2×10^{-3}.
- All the experiments are implemented in the PyTorch framework [48] with 2×Nvidia GTX 1080 Ti (except for evaluating the time cost).

4.2 Ablation Study

In this subsection, we analyze whether and how multiple scales of the channel-wise, spatial-wise attention and recursive refinement are beneficial for FGIC tasks. We use a

Table 1 Results of the ablation study

	Accuracy		Parameters	GFLOPs
	CUB-200-2011	Stanford Cars		
Baseline	75.9%	89.3%	9.327M	30.031
$\Omega_8^{chl}(.)$	81.2%	90.5%	9.393M	30.031
$\Omega_{16}^{chl}(.)$	81.4%	90.4%	9.360M	30.031
$\Omega_{32}^{chl}(.)$	81.6%	90.7%	9.343M	30.031
$\Omega_8^{chl}(.)+\Omega_{16}^{chl}(.)+\Omega_{32}^{chl}(.)$	81.8%	90.8%	9.443M	30.031
$\Omega_3^{spat}(.)$	81.7%	90.6%	9.327M	30.031
$\Omega_5^{spat}(.)$	82.2%	90.6%	9.327M	30.031
$\Omega_7^{spat}(.)$	81.6%	90.6%	9.327M	30.031
$\Omega_3^{spat}(.)+\Omega_5^{spat}(.)+\Omega_7^{spat}(.)$	81.6%	90.8%	9.327M	30.031
$\boldsymbol{\Omega}^{spat}+\boldsymbol{\Omega}^{chl}$, $T=1$	81.9%	91.3%	9.443M	30.031
$\boldsymbol{\Omega}^{spat}+\boldsymbol{\Omega}^{chl}$, $T=2$	81.9%	91.5%	9.443M	30.032
$\boldsymbol{\Omega}^{spat}+\boldsymbol{\Omega}^{chl}$, $T=3$	82.4%	92.1%	9.443M	30.032
$\boldsymbol{\Omega}^{spat}+\boldsymbol{\Omega}^{chl}$, $T=4$	81.4%	91.9%	9.443M	30.032
$\boldsymbol{\Omega}^{spat}+\boldsymbol{\Omega}^{chl}$, $T=5$	80.9%	91.6%	9.443M	30.032

Table 2 Comparison results with baselines

	Accuracy		Parameters	GFLOPs	Time Cost
	CUB-200-2011	Stanford Cars			
VGG11_bn	75.9%	89.3%	9.327M	30.031	49.209ms
VGG11_bn+RMCSAM	82.4%	92.1%	9.443M	30.032	52.102ms
VGG16_bn	80.1%	91.9%	14.824M	61.549	97.108ms
VGG16_bn+RMCSAM	83.6%	93.0%	14.940M	61.550	98.173ms
Resnet18	79.9%	92.1%	11.277M	7.274	16.181ms
Resnet18+RMCSAM	80.5%	92.9%	11.394M	7.275	20.152ms
Resnet50	85.5%	93.2%	23.910M	16.438	48.000ms
Resnet50+RMCSAM	86.1%	94.2%	25.751M	16.449	54.100ms
Gluon_resnet18_v1b	81.9%	92.6%	11.277M	7.274	15.938ms
Gluon_resnet18_v1b+RMCSAM	82.7%	93.0%	11.394M	7.275	19.339ms
GoogLeNet	80.5%	93.4%	5.801M	6.016	25.126ms
GoogLeNet+RMCSAM	80.9%	93.8%	6.263M	6.018	29.023ms

VGG11 network [29] with batch normalization [45] as the baseline, and evaluate the performance of: the baseline, the baseline + different single-scale channel-wise attention modules, the baseline + multi-scale channel-wise attention module, the baseline + different single-scale spatial-wise attention modules, the baseline + multi-scale spatial-wise attention module, the baseline + RMCSAM respectively refined 1∼5 times.

The ablation study is conducted on both datasets, and the results are shown in Table 1.

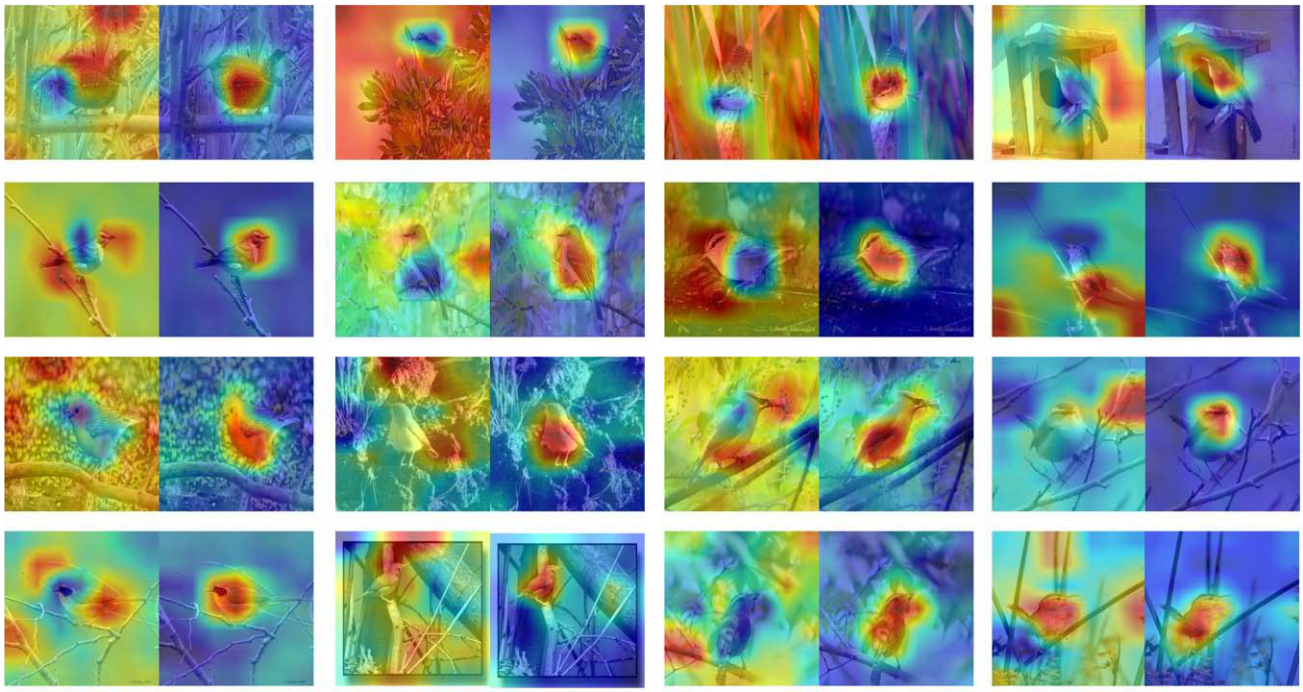
Single-scale attention vs. multi-scale attention. On both datasets, the multi-scale channel-wise attention module performs better than all the single-scale channel-wise attention modules. Compared with the baseline, the multi-scale channel-wise attention module improves the accuracy by 5.9% on CUB-200-2011 and 1.5% on Stanford Cars. Multi-scale spatial-wise attention module performs better than all the single-scale spatial-wise attention modules. Compared with the baseline, the multi-scale spatial-wise attention module improves the accuracy by 5.7% on CUB-200-2011 and 1.5% on Stanford Cars.

The influence of refining times. Simply Aggregating both multi-scale channel-wise and spatial-wise attention (i.e., $\boldsymbol{\Omega}^{spat}(.)+\boldsymbol{\Omega}^{chl}(.)$ with $T=1$) performs better than only using one of them, which suggests multi-scale channel-wise and spatial-wise attention are complementary to each other. Moreover, increasing refining times can further affect the accuracy. On both two datasets, the most suitable T is 3, because the accuracy tends to decrease with a T larger than 3. Compared with the baseline, by setting T as 3, RMCSAM improves the classification accuracy by 6.5% on CUB-200-2011 and 2.8% on Stanford Cars, while increasing only 0.116M parameters and 0.001 GFLOPs.

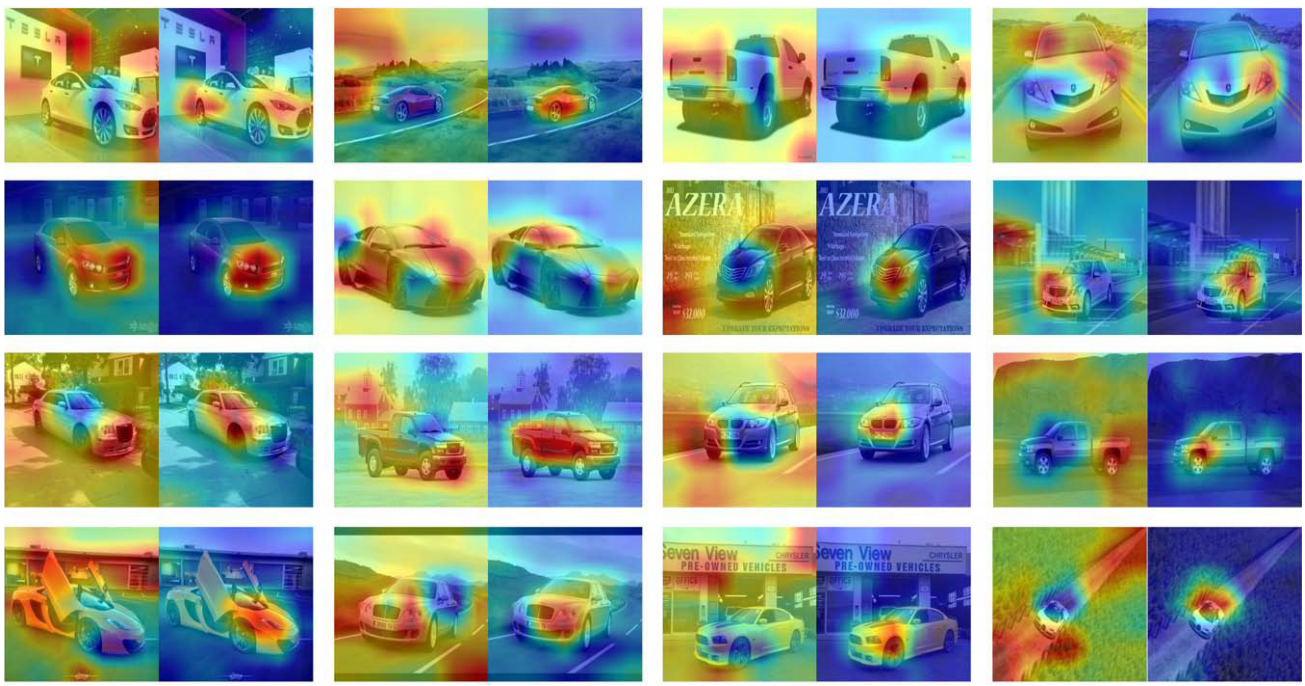
For all the rest experiments, the T for RMCSAM is set as 3.

4.3 Comparison with the Baselines

In this subsection, we empirically show how RMCSAM helps improve the classification accuracy over different baseline networks. We use as baselines six network models, namely VGG11 [29] with batch normalization, VGG16 [29] with batch normalization, Resnet18 [28], Resnet50 [28], Gluon_resnet18_v1b [49], and GoogLeNet [50]. We compare the networks with and without the proposed module, and the results are shown in Table 2. RMCSAM favorably improves the classification of all the baselines by 0.4%∼6.5% on CUB-200-2011 and 0.4%∼2.8% on Stanford Cars. In terms of the extra overhead, RMCSAM increases only 0.116M∼1.841M parameters and 0.001∼0.003 GFLOPs. In view of the negligible additional parameters and GFLOPs, our approach provides a good improvement in classification accuracy. Regarding the additional time cost, RMCSAM increases 1.065ms∼6.100ms over different back-



(a) CUB-200-2011



(b) Stanford Cars

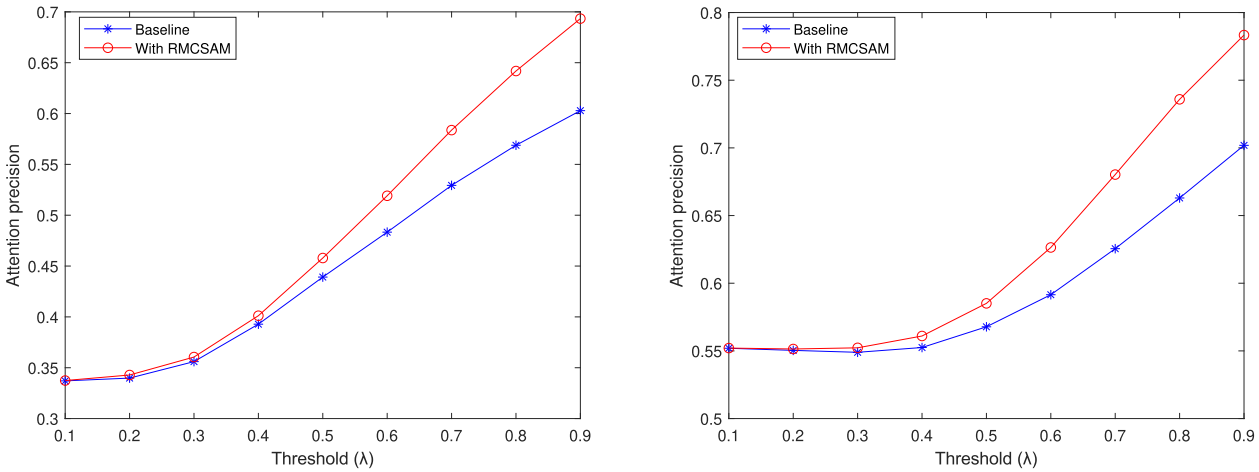
Fig. 4 Visualization of Grad-CAM. In each pair of images, the left one is the visualization results using the baseline network. The right one is the visualization results using the network inserted with RMCSAM.

bones for processing a group of eight input images, which is also a small overhead.

4.4 Analysis of Attention Capturing

In this subsection, we evaluate whether the proposed RMCSAM actually helps a network focus on discriminative visual information by two methods, namely visualization and quantitative analysis. The experiments in this subsection are implemented with the VGG11 model with batch normalization.

First, we use Grad-CAM [51] to visualize the focus of



(a) CUB-200-2011 (b) Stanford Cars

Fig. 5 Attention precision with different thresholds

the networks. Grad-CAM uses the gradients of the predicted category, flowing into the final convolutional layer to generate a heatmap highlighting the important regions in the image for predicting the category. That is, the heatmap generated by Grad-CAM visualizes the "reason" why the network "thinks" a given image belongs to a certain category. The visualization results are shown in Fig. 4. Compared with the baseline network, the network inserted with RMCSAM focuses more on discriminative regions and objects.

Second, we quantitatively analyze the attention capturing ability by attention precision. We first introduce the definition of attention precision. The computation of attention precision starts from generating a heatmap $Y \in \mathbb{R}^{H' \times W'}$ by Grad-CAM, which has the same spatial size as the input image ($\mathbb{R}^{H' \times W' \times 3}$). Regard Y as a set of pixels, namely $Y = \{y_{(1,1)}, y_{(1,2)}, \ldots, y_{(\alpha,\beta)}, \ldots y_{(H',W')}\}$. Then Y is normalized as:

$$y'_{(\alpha,\beta)} = \frac{y_{(\alpha,\beta)} - min(Y)}{max(Y) - min(Y)} \tag{15}$$

After the normalization, each value of the heat map ranges from 0 to 1. Then given a threshold λ ($0 < \lambda < 1$), all the values larger than λ are set as 1, and all the values no larger than λ are set as 0 as:

$$y''_{(\alpha,\beta)} = \begin{cases} 1, & if\ y'_{(\alpha,\beta)} - \lambda > 0 \\ 0, & if\ y'_{(\alpha,\beta)} - \lambda <= 0. \end{cases} \tag{16}$$

Thereafter, the attention precision AP is given as:

$$AP = \frac{N_{in}}{N_{in} + N_{out}}, \tag{17}$$

where N_{in} denotes the total number of pixels locating inside the manually labeled bounding box and having a value of 1. N_{out} denotes the total number of pixels locating outside the manually labeled bounding box and having a value of 1. The manually labeled bounding boxes are officially provided by the authors of the two datasets [30], [31]. The bounding boxes are widely used as the ground truth in fine-grained object detection or segmentation tasks [16], [52], [53].

The attention precision expresses the proportion of the pixels the networks "consider" to be discriminative actually are discriminative. We evaluate the attention precision with different thresholds of 0.1~0.9. The results are shown in Fig. 5. Overall, the network inserted with RMCSAM has much higher attention precision than the baseline. With the increase of λ, the gap of attention precision between them is getting wider and wider. A higher threshold selects the pixels that have more contribution to the final prediction. That is, the network inserted with RMCSAM tends to "consider" a higher proportion of pixels inside the bounding box as high-contribution pixels than the baseline.

4.5 Comparison with the State-of-the-Art Attention Modules in Fine-Grained Image Classification Task

In this subsection, we compare our proposed module with other state-of-the-art attention modules in FGIC tasks. We adopt Resnet50 as the backbone because it is the most commonly used network backbone for analyzing the performance of attention modules [24]–[27]. The results are shown in Table 3. The best accuracy and lowest overhead are highlighted in bold. Basically, the proposed attention module outperforms the other ones in terms of classification accuracy. The SE module [24], CBAM [27], and BAM [26] require lower overhead than our proposed module, but the accuracy of our proposed module is clearly higher than theirs on both datasets. AFF [25] has the closest classification accuracy to ours on both datasets but requires a little more time cost and much more GFLOPs and parameters.

4.6 Comparison with the Previous Approaches in Fine-grained Image Classification Task

In this subsection, we compare our proposed approach with the approaches achieving state-of-the-art accuracy in FGIC

Table 3 Comparison results with state-of-the-art attention modules in FGIC task

	Accuracy		Parameters	GFLOPs	Time Cost
	CUB-200-2011	Stanford Cars			
Resnet50+SE [24]	85.2%	93.9%	**24.434M**	**16.439**	**52.487ms**
Resnet50+AFF [25]	86.1%	94.1%	32.318M	17.676	54.294ms
Resnet50+iAFF [25]	85.6%	93.8%	34.423M	18.091	58.423ms
Resnet50+DAF [25]	85.8%	93.9%	28.108M	17.261	53.701ms
Resnet50+BAM [26]	85.7%	93.8%	24.998M	16.549	53.457ms
Resnet50+CBAM [27]	85.5%	93.4%	24.436M	16.440	53.322ms
Resnet50+RMCSAM (ours)	**86.2%**	**94.2%**	25.751M	16.449	54.100ms

Table 4 Comparison results with state-of-the-art approaches in FGIC task

	Backbone	Accuracy		Parameters	GFLOPs	Time Cost
		CUB-200-2011	Stanford Cars			
GaRD [43]	Resnet50	89.6%	94.3%	23.871M	18.589	17.848ms
DAM [42]	Resnet50	87.5%	94.4%	23.508M	49.314	64.285ms
PCA-Net [33]	Resnet50	88.3%	94.3%	**21.270M**	184.317	61.202ms
TransFG [38]	ViT-B_16 [37]	**91.7%**	94.8%	85.762M	107.564	259.633ms
ACNet [41]	Resnet50	88.1%	94.6%	197.264M	155.497	184.287ms
ProtoTree [40]	Resnet50	87.2%	91.5%	24.032M	**8.270**	160.731ms
PMG [32]	Resnet50	89.6%	95.1%	45.132M	37.316	20.913ms
RMCSAM	Resnet50	86.2%	94.2%	25.751M	16.449	**13.694ms**
RMCSAM*	Resnet50	87.2%	94.7%	25.751M	16.449	**13.694ms**
RMCSAM*+PCA-Net	Resnet50	88.9%	95.0%	23.112M	184.384	72.143ms
RMCSAM*+PMG	Resnet50	89.9%	**95.3%**	46.973M	37.328	25.904ms

* $\star$ illustrates the RMCSAM that is pre-trained on the ImageNet.
** In this table, the time cost is evaluated with a single image (i.e., a $1 \times 3 \times 448 \times 448$ tensor).

tasks. We use Resnet50 as the backbone because it is most widely used in those studies [32], [33], [40]–[43]. As mentioned before, in previous subsections, we use the CNN backbones pre-trained on the ImageNet, but the parameters of the RMCSAM are initialized randomly. In this subsection, for better accuracy, we also present the experimental results by using the RMCSAM parameters pre-trained together with the Resnet50 on the ImageNet, which is marked as $\star$.

The pretraining is trained from scratch and conducted with the official Timm toolbox [54] on 2×Nvidia RTX 3080 Ti. We also train an original Resnet50 under the exact same configuration as a baseline. We turn on automatic mixed precision [55] and label smoothing [56]. We set the batch size as 256 and train the networks using standard Stochastic Gradient Descent (SGD) with the momentum of 0.9. We totally train the networks on the ImageNet for 180 epochs. Regarding the learning rate schedule, we divide the 180 epochs into 6×30 epochs. For the first 30 epochs, we train the Resnet50 with/without the RMCSAM by the constant learning rate of 0.1 for the quick decrease of training loss. From the second 30 epochs, we train the networks using cosine annealing [57], and the starting learning rate for the second 30 epochs is 0.05. Then, for every 30 epochs, we restart the cosine annealing schedule and decrease the starting learning rate by 0.7. The training of the baseline Resnet50 and the Resnet50 inserted with RMCSAM is conducted once. With RMCSAM, the average accuracy of the last 10 epochs on the validation set of the ImageNet is improved from 77.7% to 78.5%. The best accuracy of the whole 180 epochs on the validation set of the ImageNet is improved from 78.1% to 78.9%.

All the other experiments in this subsection follow the general configuration of this paper. Namely, all the other experiments in this subsection are reproduced for 10 times, and we report the average accuracy. After the pre-training, we use the weights of the pre-trained RMCSAM to replace the randomly initialized RMCSAM weights for fine-tuning on the fine-grained image classification datasets.

Moreover, as an insertable module that can improve the accuracy of the backbone CNNs, our approach intuitively looks complementary to some state-of-the-art FGIC approaches. It is possible to combine our approach with other approaches for better accuracy. Specifically, we insert the RMCSAM pre-trained on the ImageNet into the Resnet50 backbones of PMG [32] and PCA-net [33]. For a fair comparison, all the other parameters (including the parameters of the Resnet50 backbones) are initialized in the same way as the original PMG or PCA-net.

The comparison in this subsection is conducted in terms of both accuracy and computational costs. As many state-of-the-art FGIC approaches require extremely huge memory, such as [38], we test the time cost by processing one 448×448 image (i.e., a $1 \times 3 \times 448 \times 448$ tensor) to prevent the out-of-memory exception in this subsection. The comparison results are shown in Table 4. The best accuracy and lowest overhead are highlighted in bold. With the RMCSAM pre-trained on the ImageNet and Resnet50 backbone, the accuracy of our approach is very close to the state-of-the-art accuracy on the Stanford Cars and a little behind the state-of-the-art accuracy on the CUB-200-2011. TransFG achieves the best accuracy on the CUB-200-

2011 but requires huge computational overhead regarding the parameters, GFLOPs, and time cost. In contrast, our approach requires much less overhead. Especially, our approach requires 13.694ms for processing a single image at once, which is the least time cost among the approaches and around 5.3% of the time cost of TransFG. Besides, our approach has the similar accuracy as TransFG on the Stanford Cars.

Among the approaches, PCA-Net [33] has the fewest parameters, and ProtoTree [40] has the fewest GFLOPs. However, they require much more time cost than the proposed approach, which is caused by the complex feature extracting and aggregating framework (PCA-Net) or the tree architecture hardly parallelizable (ProtoTree). Besides, on the Stanford Cars, our approach has better accuracy than both PCA-Net and ProtoTree.

By combining with our approach, the accuracy is improved by on both datasets. Especially, RMCSAM*+PMG achieves 95.3% accuracy on Stanford Cars, which surpasses the previous best accuracy on this dataset. It achieves 89.9% accuracy on the CUB-200-2011, which surpasses the previous best accuracy obtained with Resnet50 backbone on this dataset. Among the 10 times of repeated experiments of RMCSAM*+PMG, the lowest accuracies are 89.7% (CUB-200-2011) and 95.3% (Stanford Cars), while the highest accuracies are 90.0% (CUB-200-2011) and 95.5% (Stanford Cars). On both datasets, the highest, lowest and average accuracies of RMCSAM*+PMG are better than the best accuracies reported in [32], which shows our approach can bring stable improvement over the original PMG. Considering that the accuracy of PMG, the state-of-the-art approach, is already very high, it is interesting to see there is still room for improvement by our proposed module.

5. Conclusion

We propose the recursive multi-scale channel-spatial attention module (RMCSAM), a new approach for capturing attentional information in fine-grained image classification (FGIC) tasks. RMCSAM is designed by following the previous experience that localizing multi-scale attention regions is very effective for FGIC. However, instead of region localizing strategy, RMCSAM is designed as an insertable attention module, which can capture channel-wise and spatial-wise attention of multiple scales and accordingly refine the deep feature maps to better correspond to the visual attention. The feature maps are recursively refined a predetermined number of times to obtain the finer feature map. In this way, RMCSAM requires a very small additional overhead. The experimental results show that the multi-scale channel-wise and spatial-wise attention are complementary, and aggregation of them brings better performance. Besides, the recursive refinement can further improve the accuracy. The experimental results also show that RMCSAM can improve the classification accuracy of widely used network backbones and is able to improve the attention capturing ability. RMCSAM also outperforms other attention modules in FGIC tasks. Moreover, our approach can be combined with PMG and PCA-Net framework, which are state-of-the-art approaches in the FGIC task, to further improve the accuracy.

References

[1] W. Ge, X. Lin, and Y. Yu, "Weakly supervised complementary parts models for fine-grained image classification from the bottom up," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.3034–3043, 2019.
[2] Y. Ding, Z. Ma, S. Wen, J. Xie, D. Chang, Z. Si, M. Wu, and H. Ling, "Ap-cnn: weakly supervised attention pyramid convolutional neural network for fine-grained visual classification," IEEE Transactions on Image Processing, vol.30, pp.2826–2836, 2021.
[3] T. Rao, X. Li, H. Zhang, and M. Xu, "Multi-level region-based convolutional neural network for image emotion classification," Neurocomputing, vol.333, pp.429–439, 2019.
[4] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.5012–5021, 2019.
[5] X. He, Y. Peng, and J. Zhao, "Fast fine-grained image classification via weakly supervised discriminative localization," IEEE Transactions on Circuits and Systems for Video Technology, vol.29, no.5, pp.1394–1407, 2018.
[6] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, "Learning to navigate for fine-grained classification," Proceedings of the European Conference on Computer Vision, vol.11218, pp.438–454, 2018.
[7] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.4438–4446, 2017.
[8] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M.N. Do, "Weakly supervised fine-grained categorization with part-based image representation," IEEE Transactions on Image Processing, vol.25, no.4, pp.1713–1725, 2016.
[9] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," Proceedings of the IEEE international conference on computer vision, pp.1143–1151, 2015.
[10] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.842–850, 2015.
[11] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," Proceedings of the IEEE international conference on computer vision, pp.1641–1648, 2013.
[12] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked cnn for fine-grained visual categorization," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.1173–1182, 2016.
[13] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep lac: Deep localization, alignment and classification for fine-grained recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.1666–1674, 2015.
[14] O.M. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman, "The truth about cats and dogs," 2011 International Conference on Computer Vision, pp.1427–1434, IEEE, 2011.
[15] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas, "Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

pp.1143–1152, 2016.
[16] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," European conference on computer vision, vol.8689, pp.834–849, Springer, 2014.
[17] M. Jaderberg, K. Simonyan, A. Zisserman, et al., "Spatial transformer networks," Advances in neural information processing systems, vol.28, pp.2017–2025, 2015.
[18] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," IEEE Transactions on Image Processing, vol.27, no.3, pp.1487–1500, 2017.
[19] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.1134–1142, 2016.
[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," Proceedings of the IEEE international conference on computer vision, pp.2961–2969, 2017.
[21] C. Sutton and A. McCallum, "An introduction to conditional random fields," Mach. Learn, vol.4, no.4, pp.267–373, 2012.
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol.9, no.8, pp.1735–1780, 1997.
[23] H. Zhao, Y. Zhang, S. Liu, J. Shi, C.C. Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," Proceedings of the European Conference on Computer Vision (ECCV), vol.11213, pp.270–286, 2018.
[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7132–7141, 2018.
[25] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp.3560–3569, 2021.
[26] J. Park, S. Woo, J.-Y. Lee, and I.S. Kweon, "A simple and lightweight attention module for convolutional neural networks," International Journal of Computer Vision, vol.128, no.4, pp.783–798, 2020.
[27] S. Woo, J. Park, J.-Y. Lee, and I.S. Kweon, "Cbam: Convolutional block attention module," Proceedings of the European conference on computer vision (ECCV), vol.11211, pp.3–19, 2018.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.
[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[30] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.
[31] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.
[32] R. Du, D. Chang, A.K. Bhunia, J. Xie, Z. Ma, Y.-Z. Song, and J. Guo, "Fine-grained visual classification via progressive multi-granularity training of jigsaw patches," European Conference on Computer Vision, vol.12365, pp.153–168, Springer, 2020.
[33] T. Zhang, D. Chang, Z. Ma, and J. Guo, "Progressive co-attention network for fine-grained visual classification," arXiv preprint arXiv:2101.08527, 2021.
[34] T.H. Kim, M.S. Sajjadi, M. Hirsch, and B. Scholkopf, "Spatio-temporal transformer network for video restoration," Proceedings of the European Conference on Computer Vision (ECCV), vol.11207, pp.111–127, 2018.
[35] K.M. Schatz, E. Quintanilla, S. Vyas, and Y.S. Rawat, "A recurrent transformer network for novel view action synthesis," Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, Aug. 23–28, 2020, Proceedings, Part XXVII 16, vol.12372, pp.410–426, Springer, 2020.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, pp.5998–6008, 2017.
[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.
[38] J. He, J.N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, and A. Yuille, "Transfg: A transformer architecture for fine-grained recognition," arXiv preprint arXiv:2103.07976, 2021.
[39] S.R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE transactions on systems, man, and cybernetics, vol.21, no.3, pp.660–674, 1991.
[40] M. Nauta, R. van Bree, and C. Seifert, "Neural prototype trees for interpretable fine-grained image recognition," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.14933–14943, 2021.
[41] R. Ji, L. Wen, L. Zhang, D. Du, Y. Wu, C. Zhao, X. Liu, and F. Huang, "Attention convolutional binary neural tree for fine-grained visual categorization," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10468–10477, 2020.
[42] F. Xu, M. Wang, W. Zhang, Y. Cheng, and W. Chu, "Discrimination-aware mechanism for fine-grained representation learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.813–822, 2021.
[43] Y. Zhao, K. Yan, F. Huang, and J. Li, "Graph-based high-order relation discovery for fine-grained recognition," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.15079–15088, 2021.
[44] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp.248–255, 2009.
[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International conference on machine learning, pp.448–456, PMLR, 2015.
[46] Q. Xu and L. Zhang, "The effect of different hidden unit number of sparse autoencoder," The 27th Chinese Control and Decision Conference (2015 CCDC), pp.2464–2467, 2015.
[47] M.D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," European conference on computer vision, vol.8689, pp.818–833, Springer, 2014.
[48] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
[49] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.558–567, 2019.
[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.1–9, 2015.
[51] R.R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that?," arXiv preprint arXiv:1611.07450, 2016.
[52] B. Zhao, J. Feng, X. Wu, and S. Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," International Journal of Automation and Computing, vol.14, no.2, pp.119–135, 2017.
[53] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.811–818, 2013.
[54] R. Wightman, "Pytorch image models." https://github.com/rwightman/pytorch-image-models, 2019.
[55] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D.

Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, et al., "Mixed precision training," arXiv preprint arXiv:1710.03740, 2017.

[56] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?," arXiv preprint arXiv:1906.02629, 2019.

[57] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," 5th International Conference on Learning Representations, 2017.

Dichao Liu received the M.S. degree in Information Science, from Nagoya University in 2018. He is currently a Ph.D. Candidate with the Graduate School of Informatics, Nagoya University. His current research interests include fine-grained image classification and fine-grained human action recognition.

Yu Wang received the M.S. degree in Information Science and Ph.D. degree in Engineering, from Nagoya University, in 2010 and 2013, respectively. He is currently an assistant professor with the College of Information Science and Engineering, Ritsumeikan University. His current research interests include image Recognition, video analysis, object detection, and deep learning.

Kenji Mase received B.E. degree in Electrical Engineering and M.E. and Ph.D. degrees in Information Engineering from Nagoya University in 1979, 1981 and 1992, respectively. He became professor of Nagoya University in August 2002. He is now with the Graduate School of Informatics, Nagoya University. He is Research Supervisor of JST CREST on Symbiotic Interactions since 2017. He joined the Nippon Telegraph and Telephone Corporation NTT in 1981 and had been with the NTT Human Interface Laboratories. He was a visiting researcher at the Media Laboratory, MIT in 1988-1989. He has been with ATR (Advanced Telecommunications Research Institute) in 1995-2002. His research interests include gesture recognition, computer graphics, artificial intelligence and their applications for computer-aided communications, wearable/ubiquitous computers and lifelog. He is a Fellow of Institutes of Electronics, Information and Communication Engineers (IEICE) of Japan, and member of the Information Processing Society of Japan (IPSJ), Japan Society of Artificial Intelligence (JSAI), Virtual Reality Society of Japan, Human Interface Society of Japan and ACM, and a senior member of IEEE Computer Society. He was a Section Chair of IEEE Nagoya Section in 2014-2015. He is the 24th -25th associate member of Science Council of Japan.

Jien Kato received the M.E. and Ph.D. degrees in Information Engineering from Nagoya University in 1990 and 1993, respectively. Then she became an assistant professor at Toyama University. She was a visiting researcher at the University of Oxford from 1999 for one year. She became an associate professor at the Graduate School of Engineering of Nagoya University in 2000. She has been a professor at the College of Information Science and Engineering of Ritsumeikan University since 2018. Her research interests include object recognition, visual event recognition and machine learning. She is a member of IEICE, IPSJ and JSAI, and also a senior member of IEEE. Her current research interests include image Recognition, video analysis, object detection, and deep learning.