Pattern Recognition 156 (2024) 110849

Contents lists available at ScienceDirect

Pattern Recognition

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

# Fine-grained recognition via submodular optimization regulated progressive training

Bin Kang [a], Songlin Du [b,*], Dong Liang [c], Fan Wu [a], Xin Li [a]

[a] College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China
[b] School of Automation, Southeast University, Nanjing, China
[c] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

ARTICLE INFO

*Keywords:*
Fine-grained recognition
Progressive training
Submodular optimization

ABSTRACT

Progressive training has unfolded its superiority on a wide range of downstream tasks. However, it may fail in fine-grained recognition (FGR) due to special challenges with high intra-class and low inter-class variances. In this paper, we propose an active self-pace learning method to exploit the full potential of progressive training strategy in FGR. The key innovation of our design is to integrate submodular optimization and self-pace learning into a maximum–minimum optimization framework. The submodular optimization is regarded as a dynamic regularization to select active sample groups in each training round for restricting the search space of self-pace optimization. This can overcome the limitation of traditional self-pace learning that is easily trapped into local minimums when facing challenging samples. Extensive experiments on three public FGR datasets show that the proposed method can win at least 1.5% performance gain in various kinds of network backbones including swin-transformer.

## 1. Introduction

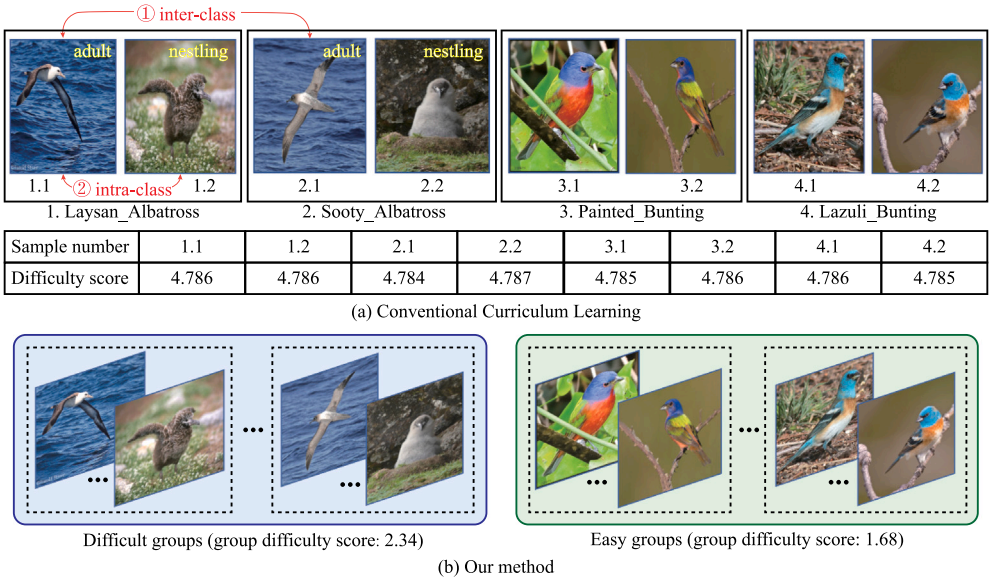
Fine-grained recognition (FGR) has been highly regarded in recent years due to its practical purpose to distinguish between similar subcategories. Related works have been successfully applied to intelligent driving and retail product, etc. Different from traditional image recognition tasks, FGR requires the classification model to discriminate subtle difference with small clue [1–4]. It is a tough problem because the natural occlusion and observation angle may cover the implicit but key target regions. With the rapid development of deep learning technology, FGR has made significant progress. This gives credit to the design of network structure that can simultaneously locate and represent the discriminative target sub-regions by image labels. In practical terms, apart from the network structures, a well-designed progressive training strategy also can significantly enhance the accuracy of the FGR model. Unfortunately, few works have done it well.

Curriculum learning (CL) is the representative method for progressive training. A typical framework of curriculum learning is composed of two separated components [5]: the difficulty measurer and the training scheduler. Difficulty measurer is aimed to employ the difficulty function to rank training samples. While training scheduler focuses on designing the training strategies to divide ranking results into appropriate training batches. Introducing curriculum learning in FGR will be conducive to the location of target details because the process of ordering training samples can encourage the network to locate discriminative information from coarse to fine-grained level. This is actually to imitate the child's behavior in the image cognition course, where children are asked to focus on the image details progressively. Although CL is a promising work in FGR, it will face special challenges such as shown in Fig. 1, e.g. intra-class samples may contain targets in various growth stages. The appearance of adults and nestlings is totally different. By contrast, differences between similar classes may be subtle. In this case, FGR involves high intra-class variance and small inter-class variance. Traditional curriculum learning methods quantify each sample as an independent entity. If they are adopted directly in FGR, the difficulty scores of different samples are very close (see Fig. 1(a)). This indicates that treating samples as independent entities cannot fully encourage the superiority of CL. Few curriculum learning works have been successfully employed in FGVC task. Although some works [6–8] study the progressive training of FGVC, their focus is the training strategies, not including ranking the training samples.

In fact, examples such as "adult-nestling birds" and "Laysan-Sooty" have not existed all over the category. This means that the inter and intra relations among category subsets are varied. Based on this observation, we do not distinguish the difficulty of individual samples.

* Corresponding author.
*E-mail addresses:* kangb@njupt.edu.cn (B. Kang), sdu@seu.edu.cn (S. Du), liangdong@nuaa.edu.cn (D. Liang), 1220076505@njupt.edu.cn (F. Wu), xinli@njupt.edu.cn (X. Li).

https://doi.org/10.1016/j.patcog.2024.110849
Received 1 August 2023; Received in revised form 25 July 2024; Accepted 25 July 2024
Available online 27 July 2024




| Sample number | 1.1 | 1.2 | 2.1 | 2.2 | 3.1 | 3.2 | 4.1 | 4.2 |
|---|---|---|---|---|---|---|---|---|
| Difficulty score | 4.786 | 4.786 | 4.784 | 4.787 | 4.785 | 4.786 | 4.786 | 4.785 |

(a) Conventional Curriculum Learning



Difficult groups (group difficulty score: 2.34) Easy groups (group difficulty score: 1.68)

(b) Our method

**Fig. 1.** Comparing the difference between the traditional course learning method and our method.

Instead, we integrate multiple category subsets together as the basic group to quantify group difficulty. According to group difficulty, we propose an active self-pace learning method for FGR. The central insight comes from an intuition: progressive training is actually the process that iteratively selects active samples to linearly decreases the uncertainty of category prediction. If the combination of category subsets with irregular sample variance can be quantified, the process of category combination is submodular. Exploiting the submodularity can provide a near-optimal yet economic solution to rank groups from easy to hard for active sample selection. Following above discussion, we model the combination of category subsets as a submodular optimization problem. From Fig. 1(b), we can see submodular optimization is able to properly quantify the difficulty levels of category subsets. Those samples within easy groups are more active than hard groups. Under our problem formulation, we build a collaborated maximum–minimum optimization framework that integrates self-pace learning and submodular optimization into a close union. The main contributions of the proposed method are listed as follows:

(1) We thoroughly analyze the limitations of traditional curriculum learning in FGR and propose a submodular optimization model for ranking category subsets. The key innovation of our model relies on that we are the first to exploit the submodularity for active sample selection. Based on our problem formulation, the optimal category subsets can be progressively selected to obtain steady cumulative gain.

(2) To effectively train the FGR networks, we combine submodular optimization with self-pace learning to generate a collaborated maximum–minimum optimization framework. The constructed framework can achieve smooth and stable progressive learning through using active samples to restrict the search space of self-pace optimization.

(3) The proposed collaborated optimization framework can be deployed on various types of FGR networks. Extensive experiments on three fine-grained recognition datasets can verify the superiority of progressive training, where the averaged recognition gain surpasses 1.5%.

It is worth mentioning that in our previous work [9], we are the first to introduce submodular optimization in progressive training. Moreover, we also verify that the submodular optimization is robust to the challenge of FGR. The main differences between this paper and [9] are summarized as follows: Firstly, we introduce submodular optimization into self-pace learning process to make full advantage of both optimizations. In comparison, the work in [9] only uses submodular optimization to achieve progressive training. Secondly, since a collaborated optimization problem is formulated in this paper, we propose an iterative conditional sampling method to solve the problem. Finally, we conduct more extensive experiments in this paper to verify the generality of our method. It should be noted that we consider the optimization model formulated in the conference version as a dynamic regularization in the proposed collaborated optimization framework of this journal version. The main advantage of this design is that we introduce submodular optimization in the process of training process. We do not use submodular optimization to predefine the difficulty of different sub-classes as conference version. The proposed method in this paper is actually to integrate difficulty measurer and training scheduler into a joint optimization framework.

## 2. Related work

### 2.1. Fine-grained recognition

The current FGR is weakly supervised. Relevant research mainly falls into attention-based [10–12], sampling-based [13], graph-based [14–16], and transformer based methods [17]. The ACNet model [18] was a classic attention-based model that realized feature extraction and classification through an attention-based convolutional binary tree structure. This can force the model to capture multi-scale fine-grained features. In those sampling-based methods, the S3N network [19] captured peaks (local maxima) by sampling from the class response map for contextual information. The GaRD model [20] can effectively adopt a graph to obtain high-order contextual. The TransFG model [21] was the first work to apply the Transformer model to FGR. This model can use a part selection module to select image blocks containing fine-grained information, enhancing the accuracy gain. Aforementioned SOTA methods focus on network design, few works seriously study the role of the training strategy.

### 2.2. Curriculum learning

The automatic method is the research focus in current course learning. Related works can be roughly divided into three categories: self-pace learning based methods, teacher transfer-based methods and teacher reinforcement-based methods. Self-pace learning-based methods aim to let the model measure the difficulty of the training samples according to their own loss. For example, the work [22] used the concept of “contrast course” to divide the training samples into multiple

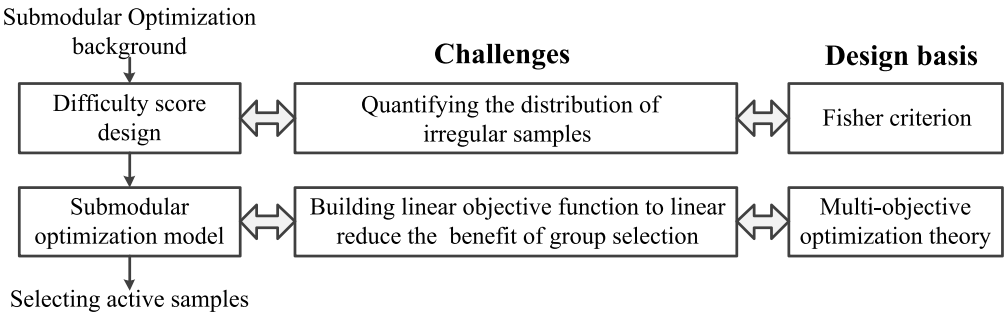



**Fig. 2.** Illustrating problem formulation on submodular optimization based category grouping.

stages. The work [23] distinguished the difficulty of samples according to the accuracy of centerline detection. Teacher transfer-based methods mainly use pre-trained models to measure the difficulty of samples. Representative works such as [24] defined the network to be trained as a student network. The confidence of student network samples was evaluated by the pre-trained teacher network combined with transfer learning strategy. The teacher reinforcement-based methods regard the network to be trained as a teacher, and performs dynamic data selection according to the feedback of students. Representative works such as [25] made the teacher network gradually adapt to more complex tasks with the assistance of feedback from the student network. In addition to the aforementioned works, methods such as Bayesian optimization, meta-learning, and hypernetworks are also introduced in the course learning to assist in the definition of difficult samples [26].

### 2.3. Submodular optimization

Submodular optimization problems require different constraints according to different scenarios. According to the constraints, submodular optimization can be divided into: cardinality constraint, knapsack constraint, and matroid constraint solutions. In cardinality constraints, the representative work [27] extended the maximization of the submodular function into a monotone submodular maximization problem for increasing the search space and convergence speed. The submodular maximization under the knapsack constraint requires finding the largest subset that satisfies the constraint. The representative work [28] used the gradient descent method to maximize the monotone submodular function. This can guarantee the theoretical upper bound of the approximation ratio. Literature [29] proposed a constrained submodule maximization algorithm GLS (Greedy Local Search). The algorithm solved the constrained submodular maximization problem by decomposing the constrained set into multiple small feasible subsets. Submodular maximization under matroid constraints maximized the value of the submodular function under the greedy process [30].

## 3. Submodular optimization based category grouping

As it is shown in Fig. 2, the problem formulation on submodular optimization based category grouping contains two steps: designing the difficulty score firstly and then building the submodular optimization model. Next, we will give a detailed discussion of two steps.

### 3.1. Submodular optimization background

Submodular optimization is a combinatorial optimization theory that utilizes submodularity for subset selection. Submodularity is defined as follows: given a finite set $\mathbf{V} = \{1, 2, \ldots, n\}$, a subset selection function $f$ is submodular if the marginal gain decreases as the subset increases

$$f(A \cup \{u\}) - f(A) \geq f(B \cup \{u\}) - f(B), A \in B \in V \text{ and } u \notin B. \quad (1)$$

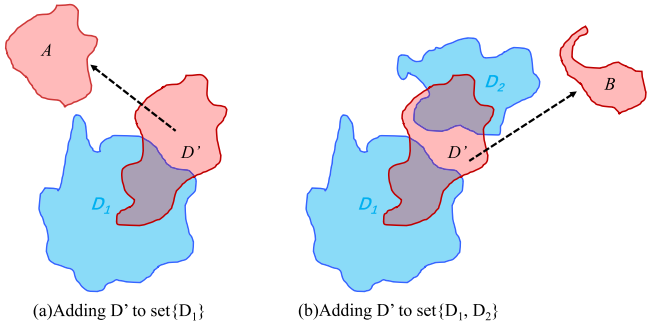



**Fig. 3.** A graphical explanation of the concept of submodularity.

Fig. 3 explains the meaning of Eq. (1). In Fig. 3(a), if a subset $D'$ is added to the set $\{D_1\}$, the enlarged area (excluding the overlapping parts) can be represented as $A$. As a comparison, in Fig. 3(b), if we add $D'$ to the set $\{D_1, D_2\}$, the enlarged area $B$ is obviously smaller than $A$. Based on the above explanation, submodularity means: the benefits generated by new subset (e.g., the newly added area in the figure) linearly decrease as the number of subsets increases. Submodular optimization can use submodularity to rank the subset combination results by evaluating the cumulative benefit of a series of candidates. In general, the submodular optimization with a cardinality parameter $k$ (i.e., number of subsets) is formulated as Eq. (2), where $D$ represents the selected subset

$$\max_{D \subset V, |D|=k} f(D). \quad (2)$$

### 3.2. Difficulty score

The difficulty score is the basis to design the subset selection function $f$ in submodular optimization theory. The group with low difficulty score means samples within this group are more active than hard groups. The definition of difficulty score involves challenging factors such as target pose variation and severe occlusion/background clutter. In addition to traditional challenging factors, there exist additional challenging factors: large intra-class variance and small inter-class variance in FGR. Large intra-class variance and small inter-class variance in category subsets imply an irregular sample variance. The Fisher criterion is a classic indicator that can be used to quantify the uncertainty of recognition accuracy according to the distribution of intra-class and inter-class variances. Inspired by this idea, we design a difficulty score, namely Group Difficulty Indicator ($GDI$) for evaluating the combination of category subsets. The designed $GDI$ is described as follows:

$$GDI = \frac{Ave\_inter}{Ave\_intra}. \quad (3)$$

The numerator $Ave\_inter$, is defined as the average inter-class similarity between a particular class and other classes. While the denominator $Ave\_intra$, is defined as the average intra-class similarity within

a particular class. Choosing $Ave\_inter$ as the numerator means that if there is significant inter-class variation between certain category subsets, a higher $Ave\_inter$ value can indicate a higher $GDI$ score. Similarly, if certain category subsets have low intra-class similarity, the $GDI$ score can also be increased by reducing the value of $Ave\_intra$. Based on Eq. (3), the designed difficulty score can distinguish between category subsets with large intra-class and small inter-class variance.

### 3.3. Submodular optimization based grouping

The submodular optimization model constructed based on Eq. (3) is

$$f(D,\lambda)=\max_{|D|=m}\frac{1}{\mid D\mid\cdot\mid S\mid}\sum_{\mathbf{X}_i\in D,\mathbf{X}_j\in S}h(\mathbf{X}_i,\mathbf{X}_j) -\lambda\frac{1}{\mid D\mid}\sum_{k,m\in X_i}\frac{2}{\mid\mathbf{X}_i\mid\cdot(\mid\mathbf{X}_i\mid-1)}h(\mathbf{x}_i^k,\mathbf{x}_j^m), \tag{4}$$

where $D$ denotes the set of candidate categories to be selected, and $S$ denotes the set of categories that are similar to $D$. Let $f$ be the subset selection function that meets submodularity. $\mathbf{X}_i$ denotes the $i$th category in the dataset, $\mathbf{X}_i=[\mathbf{x}_i^1,\dots,\mathbf{x}_i^k,\dots,\mathbf{x}_i^n]$, $n$ denotes the number of samples in category $\mathbf{X}_i$, $\mathbf{X}_j$ represent the $j$th category. $\mid D\mid=m$ is a cardinality constraint that selects $m$ categories. In Eq. (4), $h(\cdot)$ denotes a similarity metric function, where $h(\mathbf{X}_i,\mathbf{X}_j)$ measures the similarity between two categories, and $h(\mathbf{x}_i^k,\mathbf{x}_i^n)$ measures the similarity between two samples $\mathbf{x}_i^k$ and $\mathbf{x}_i^n$ within the $i$th category. The parameter $\lambda\geq 0$ balances the importance of intra-class and inter-class similarity. It can be seen that Eq. (4) is actually to select the category subset $D$ by optimizing the difficulty score, gradually combining category subsets according to the sample variance of intra-class and inter-class.

## 4. Collaborated optimization for progressive learning

### 4.1. The collaborated optimization framework

Inspired by multi-objective optimization theory, the collaborated maximum–minimum optimization framework is described as

$$\begin{aligned}&\min_{\mathbf{w}}\sum_{\mathbf{x}_i\in D}e^{c[\mathbf{x}_i]}\cdot R(y_i,L(\mathbf{x}_i,\mathbf{w}))+\sum_{i=1}^{d}c[\mathbf{x}_i],\\ &s.t.\max_{|D|=m}\frac{1}{\mid D\mid\cdot\mid S\mid}\sum_{\mathbf{X}_i\in D,\mathbf{X}_j\in S}h(\mathbf{X}_i,\mathbf{X}_j)-\lambda\frac{1}{\mid D\mid}\sum_{k,m\in\mathbf{X}_i}\frac{2}{\mid\mathbf{X}_i\mid\cdot(\mid\mathbf{X}_i\mid-1)}h(\mathbf{x}_i^k,\mathbf{x}_j^m)\end{aligned} \tag{5}$$

with

$$c[\mathbf{x}_i]=\begin{cases}1, & \text{if } R(y_i,L(\mathbf{x}_i,w))<\varphi\\ 0, & \text{if } otherwise\end{cases} \tag{6}$$

where $\mathbf{x}_i$ and $y_i$ indicates the $i$th sample and its corresponding label. $L(\mathbf{x}_i,\mathbf{w})$ represents the category prediction loss in the fine-grained recognition network, $R(\cdot)$ represents the MSE function, $c[\mathbf{x}_i]$ represents the indicate function indicating whether the sample is selected. The specific expression of indicate function is shown in Eq. (6). Eq. (5) actually combines self-pace learning with submodular optimization to form a collaborated optimization framework. The motivation of this design is to make full use of two optimizations.

Specifically, the advantage of self-pace learning lies in that it can progressively evaluate the training samples through using feedback of the network loss to select the next round training subsets. However, the limitations of self-pace learning are two-folds: (1) the sample selection closely relies on the category loss. This may cause overfitting because it cannot fully exploit the prior knowledge of category similarity; (2) it will involve high computational complexity due to the global search. To solve those problems, Eq. (5) defines submodular optimization as a dynamic regularization. The self-pace learning is established as the basis for arranging samples within groups. At this point, the submodular optimization can restrict a fine-grained search range to avoid the optimization of self-pace learning trapping into local minimums. Moreover, submodular optimization can also transfer the prior knowledge of closely related categories to self-pace learning.

### 4.2. Optimization method

In the collaborated optimization model (see Eq. (5)), the key step is the self-pace learning-based objective function for fine-grained network training. Alternative Conditional Sampling (ACS) is a classic iterative optimization method that keeps a set of variables fixed and solves for another set of variables in each iteration. Here, we adopt the ACS strategy to solve the optimization problem in Eq. (5). The specific updating strategy is as follows:

**Updating the subset of classified samples D**: the category subset is obtained by minimizing Eq. (4). Inspired by [31], we propose a random greedy algorithm that can achieve an approximation of $\frac{1}{e}$ for non-monotonic objective functions. The optimization function is as follows:

$$\max_D f(D,\lambda). \tag{7}$$

**Updating Weight Metric $\mathbf{c}[\mathbf{x}_i]$**: This step is obtained by calculating the partial derivative of Eq. (5), as shown in Eq. (7). Since $c[\mathbf{x}_i]\in[0,1]$, we can obtain a closed-form optimal solution for $c[\mathbf{x}_i]$, shown in Eq. (9). The meaning of this solution is that: when the training loss of the fine-grained network $L(y_i,g(\mathbf{x}_i,w))$ for the k-$th$ epoch is less than the threshold $\varphi$, it can be considered as a selectable sample. Otherwise, it should not be given priority for selection. As the model training iterates, the threshold $\varphi$ will increase.

The updating step of $\mathbf{c}[\mathbf{x}_i]$ are as follows:

$$\frac{\partial E}{\partial c[\mathbf{x}_i]}=c[\mathbf{x}_i]\cdot e^{c[\mathbf{x}_i]}\cdot .L(y_i,g(\mathbf{x}_i,\mathbf{w})) \tag{8}$$

Then

$$c[x_i]=\begin{cases}1, & \text{if } R(y_i,L(\mathbf{x}_i,\mathbf{w}))<\varphi\\ 0, & \text{if } otherwise\end{cases} \tag{9}$$

**Updating parameter w**: When the selected subset $D$ and the metric function $c[\mathbf{x}_i]$ are fixed, the classification weight $\mathbf{w}$ for fully connected layer of fine-grained recognition network can be updated by:

$$\mathbf{w}=min\,E(\mathbf{w},\varphi)=\arg\min_{\mathbf{w}}\sum_{\mathbf{x}_i\in D}e^{c[\mathbf{x}_i]}\cdot L(y_i,g(\mathbf{x}_i,\mathbf{w})). \tag{10}$$

**Algorithm 1:** Optimization algorithm for Eq. (5)

**Input:** Training dataset $N$, initial step sizes $\varphi$ and $\beta$
**Output:** Network parameters $\theta$

```
Initialize model parameters w
while the model has not converged, execute the outer loop do
    while the model has not converged, execute the inner loop do
        Initialize the set of difficult instances D_0 ← ∅;
        for i = 1 to m do
            M_i ⊂ N\D_{i-1}, M_i is a subset of size m that
              maximizes Σ_{μ∈M} h(μ ∪ D_{i-1}) − h(D_{i-1}).
            Randomly select an element u_i from M_i with a
              non-uniform distribution.
            Let D_i ← D_i + u_i
            Obtain the set of easy instances D at the current
              state.
            Update the feedback indicator c using Eq. (9).
            Update the classification weight w using Eq. (10).
        end
    end
    φ ← φ + β;
end
return θ* = θ;
```

The detailed optimization process is shown in Algorithm 1 and includes an inner loop and an outer loop. In the inner loop, a subset $D$ is obtained through

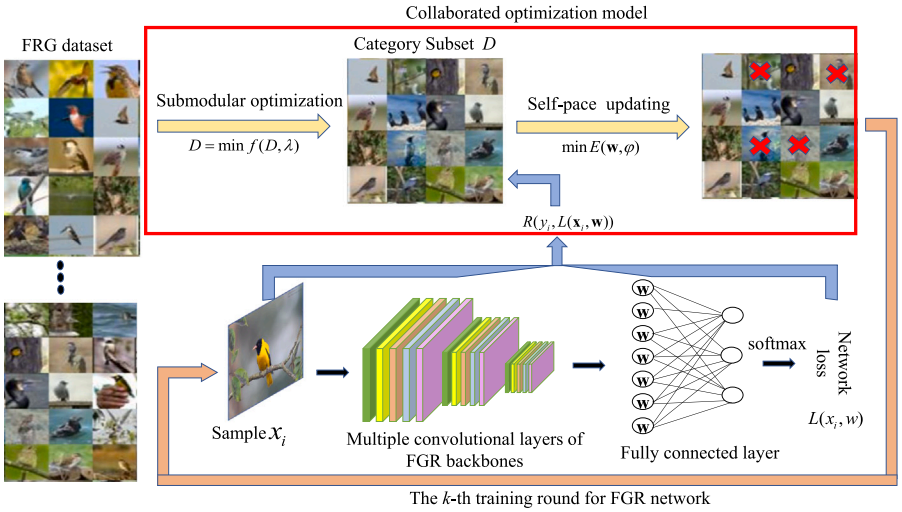



**Fig. 4.** Training framework.

a random greedy algorithm, and the variables $c$ are iteratively updated by Eq. (9). After updating $c$, the subset $D$ and variables $c$ are fixed to update the classification weight using Eq. (10). In the outer loop, more samples are gradually added to update the fine-grained backbones. In each training epoch, the classification losses are compared with a threshold, and only easily trainable samples with small loss will be reserved. As $\varphi$ increases, more samples with bigger losses (i.e., relatively difficult samples) are gradually added to train the network. The speed parameter $\varphi_u$ increases continuously through the self-increment indicator $\beta$ until the model converges.

### 4.3. Progressive training strategy

The progressive training strategy is shown in Fig. 4, where the proposed collaborated optimization model (Eq. (5)) is first to select category subset $D$ for generating image batch for the $k$th training round of FGR networks. Then, each image in subset $D$ is further evaluated through updating the self-pace optimization. This is aimed to discard those images with low contribution to the fast convergence of training loss $L(y_i, g(\mathbf{x}_i, w))$. In fact, the samples in the selected category subsets are limited, this may make self-pace optimization trapped into local minimal. To overcome this limitation, we refer to incremental learning to adopt a simple yet efficient manner: keeping the category subsets in prior training rounds during self-pace optimization. This can make the network exploit the previous knowledge to adjust the gradient. Repeating the training round, the FGR network can be progressively trained. It should be noted that, the progressive training strategy can not only be used in convolutional neural backbone, but also can be used for transformer backbones. Detailed discussion is shown in the experiments.

### 4.4. Algorithm discussion

Convergence analysis: In the procedure shown in Algorithm 1. The entire optimization is based on the ACS (Alternating Convex Search) framework. In the framework, the core optimization objective function is step 11, which is nonconvex. This step is optimized by stochastic gradient descent, which has been proven in deep learning that can give fast convergence. In addition to step 11, Eq. (5) is not a convex function. It has been proven in paper [31] that the local optimum can be reached by using a random greedy algorithm to solve the submodular optimization function, thus Eq. (5) can give convergence. Besides the above two steps, other steps are converged, thus Algorithm 1 can ensure convergence and find the local optimal solution.

Computational complexity: Here, we refer to [32] to define the time complexity of the recognition network as $O(Flops)$, where Flops means the floating-point operations of the network forward procedure. The main computational complexity of our method relies on the iteration of ACS method. In Algorithm 1, step 11 is the key point, its computational complexity is $O(nm)$, where m is the number of samples in a batch and n is the iterations. The total computational complexity is $O(n \times m \times Flops)$. Taking CUB-200-2011 dataset as an example, the iteration of Algorithm 1 is 8. Detailed experiment result please see Section 5.3.

## 5. Experimental analysis

To fully validate the effectiveness of the proposed optimization model, we carry out experiments from two aspects: quantitative analysis and qualitative analysis. First, quantitative analysis will be conducted on the CUB-200-2011 [33], Stanford Dogs [30], Stanford Cars [34] and iNaturalist 2017 [35] datasets. Qualitative analysis will be performed on the heat map during training process to show whether our progressive training can give a strong support to locate the fine-grained discriminative regions.

This analysis will be divided into three parts: evaluating the performance of the proposed method on various fine-grained recognition models, discussing the model parameters, and comparing models with and without feedback to verify the feasibility of the feedback-based model. Second, qualitative analysis will be performed on the proposed method. Through visualization of feature maps, the key features of fine-grained images will be intuitively analyzed to determine whether the proposed method can capture them effectively.

### 5.1. Experimental setup

The batch size was set to 64, and the SGD optimizer with momentum 0.9 was selected to optimize the classifier. The experiments are conducted using Python on four Nvidia Tesla P100 GPUs. In the experiment, the CUB-200-2011 dataset contains 11,788 images, of which 5994 were used for training and the rest were used for testing. The Stanford Dogs dataset contains 20,580 images, of which 12,000 images were used for training and the rest were used for testing. Similarly, The Stanford Cars dataset contains 16,182 images, 8041 images are used for testing. To ensure fair comparison, the datasets were preprocessed by resizing the input images to 600*600 and then cropping them to 448*448, which is suitable for both CUB-200-2011 and Stanford Dogs datasets. During training, random cropping was used while center cropping was used during testing. The initial learning rate was set to 0.01 for CUB-200-2011 and 0.003 for Stanford Dogs.

### 5.2. Quantitative analysis

The experimental results are shown in Table 1. From Table 1, in CUB-200-2011 datasets, it is clear that it gives a performance gain over 2% when

**Table 1**
Overall performance on three public datasets.

| Method | BaseModel | CUB-200–2011 | | Stanford Dogs | | Stanford Cars | |
|---|---|---|---|---|---|---|---|
| | | w/o (%) | w (%) | w/o (%) | w (%) | w/o (%) | w (%) |
| VGG16 [36] | – | 73.40 | **76.21** | 68.32 | **71.11** | 82.12 | **85.23** |
| ResNet50 [37] | – | 82.39 | **85.40** | 84.69 | **86.90** | 90.11 | **92.28** |
| DesNet121 [38] | – | 80.79 | **83.21** | 79.91 | **82.13** | 88.08 | **90.27** |
| CPM [39] | ResNet-50 | 86.55 | **89.64** | 79.94 | **81.25** | 90.12 | **91.29** |
| S3N [40] | ResNet-50 | 87.45 | **90.12** | 87.62 | **89.02** | 93.63 | **94.98** |
| DCL [41] | ResNet-101 | 86.82 | **88.23** | 89.10 | **91.23** | 91.89 | **92.34** |
| CIN [42] | ResNet-101 | 86.34 | **90.76** | 86.98 | **89.12** | 94.22 | **96.23** |
| PMG [43] | ResNet-50 | 88.71 | **91.23** | 88.13 | **90.89** | 94.34 | **95.89** |
| SnapMix [44] | ResNet-101 | 90.12 | **92.17** | 88.32 | **89.29** | 92.12 | **93.46** |
| SEF [45] | ResNet-101 | 87.32 | **89.23** | 88.65 | **89.04** | 93.19 | **94.67** |
| TransFG [46] | VIT-B_16 | 91.89 | **92.53** | 90.18 | **90.79** | 94.31 | **96.57** |
| ViT [47] | VIT-B_16 | 91.62 | **92.55** | 91.14 | **92.02** | 93.21 | **95.28** |
| Swin [46] | Swin-S_16 | 91.81 | **92.65** | 92.52 | **92.98** | 94.28 | **96.46** |
| SIM-Trans [48] | VIT-B_16 | 91.84 | **92.79** | 92.48 | **93.33** | 94.39 | **95.43** |
| IELT [49] | VIT-B_16 | 91.81 | **92.01** | 91.84 | **92.56** | 95.62 | **95.91** |
| ViT-NeT [50] | SwinT-B_16 | 91.60 | **92.02** | 90.03 | **90.22** | 95.00 | **95.78** |

**Table 2**
Experiments on iNaturalist 2017.

| Method | Backbone | w/o | w |
|---|---|---|---|
| TransFG | VIT-B 16 | 71.6 | 71.9 |
| SIM-Trans | VIT-B 16 | 69.9 | 70.8 |

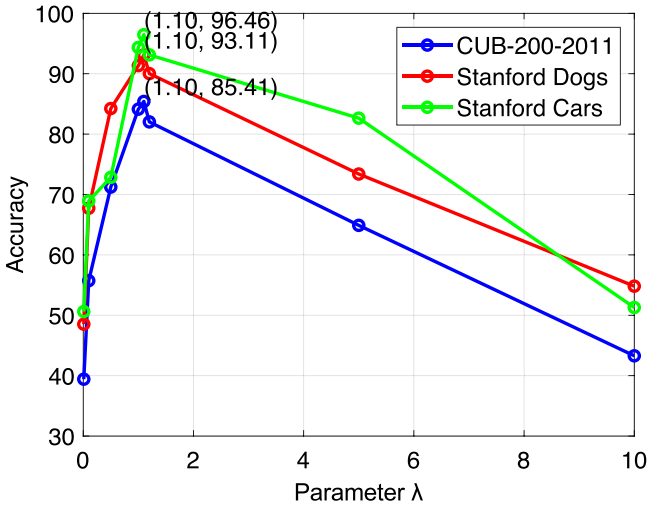



**Fig. 5.** The experiment on parameter setting. The backbone is Swin.

adding the proposed progressive training method to FGR methods. Besides FGR model, our progressive training method also gives an obvious accuracy gain (surpass 1.5%) on general network structures such as Swin and Resnet. The experiments on Stanford Dogs and Stanford Cars can also validate the superiority of our progressive training method. Especially in Stanford Cars dataset, the performance averaged accuracy gain surpass 2.5%

Here, we also carry out an experiment on iNaturalist 2017. The testing result is shown in Table 2 below. From this test we can see that our method can win 1.2% performance gain in SIM-Trans. The token selection strategy in TransFG will eliminate a large number of tokens in the challenging samples, narrowing the gap between difficult and ordinary sub-classes, thus the performance gain obtained by ranking the difficult groups is not as obvious as related works. Moreover, the comparison between previous conference and current works is shown in Table 3.

### 5.3. Efficiency of the collaborated optimization model

**Discussion on the parameter setting**: In this section, we will discuss the parameter setting of the proposed collaborated optimization model.

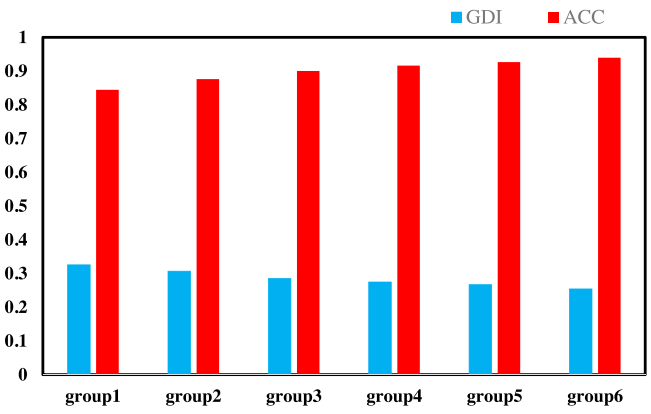



**Fig. 6.** Testing the efficiency of the difficulty score. The proposed submodular optimization is derived from group difficulty indicator (GDI). As GDI decreases, the averaged accuracy of the selected group gradually increases, indicating that the proposed submodular optimization can give a reasonable ranking result.

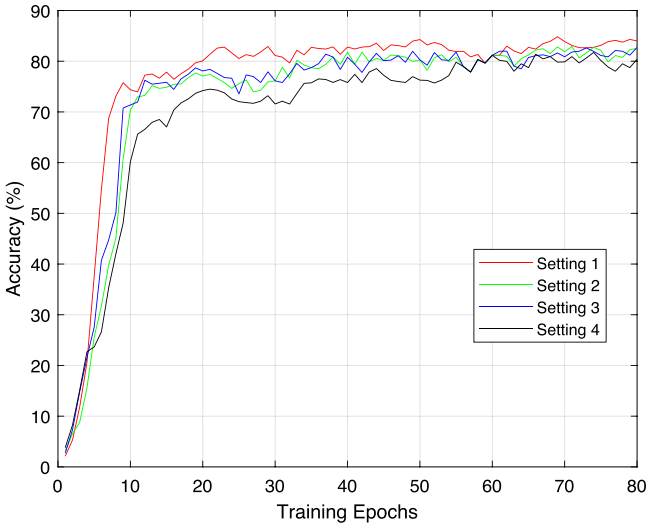



**Fig. 7.** Comparative results of ablation experiments.

Specifically, the parameter $\lambda$ is an important parameter in Eq. (5), thus we test the recognition accuracy with $\lambda$ range [0.01,10]. To give a more confident result, we carry out parameter testing on three datasets, the results are shown in Fig. 5. From Fig. 5 we could clearly see that the proposed collaborated optimization model can give the highest recognition accuracy when $\lambda$ is set as 1.1. Setting $\lambda$ too small will limit the intra-class diversity in the model, making the submodular optimization hardly obtain the optimal combination of sample subsets. On the other hand, setting $\lambda$ too big causes the model to only concentrate on intra-class diversity and ignore inter-class diversity. Three dataset gives similar results, which means parameter $\lambda$ is not sensitive to different datasets.

**Efficiency of the difficulty score**: Submodular optimization is the core in our collaborated optimization model. In this part, we will show the efficiency of the difficulty score. Specifically, it has been verified in curriculum learning that the uncertainty of image is proportional to the value of CNN loss function. This means that the accuracy of testing model can be considered as an indicator to test the ranking results. Based on this observation, we use the proposed submodular optimization to rank the groups, the pre-trained ResNet50 is used to calculate the averaged accuracy of different groups. From Fig. 6 we could clearly see that the averaged group recognition accuracy follows an increasing trend, which can verify the efficiency of group ranking result.

**Table 3**
Comparison between previous conference and current works.

| Method | CUB-200–2011 | | | Stanford Dogs | | | Stanford Cars | | |
|---|---|---|---|---|---|---|---|---|---|
| | w/o strategy(%) | w strategy(%) | | w/o strategy(%) | w strategy(%) | | w/o strategy(%) | w strategy(%) | |
| | | Conference | Journal | | Conference | Journal | | Conference | Journal |
| TransFG | 91.89 | 92.00 | 92.53 | 90.18 | 90.47 | 90.79 | 94.31 | 95.23 | 96.57 |
| Vit | 91.62 | 91.82 | 92.55 | 91.14 | 91.31 | 92.02 | 93.21 | 94.35 | 95.28 |
| Swin | 91.81 | 92.06 | 92.65 | 92.52 | 92.38 | 92.98 | 94.28 | 94.29 | 96.46 |

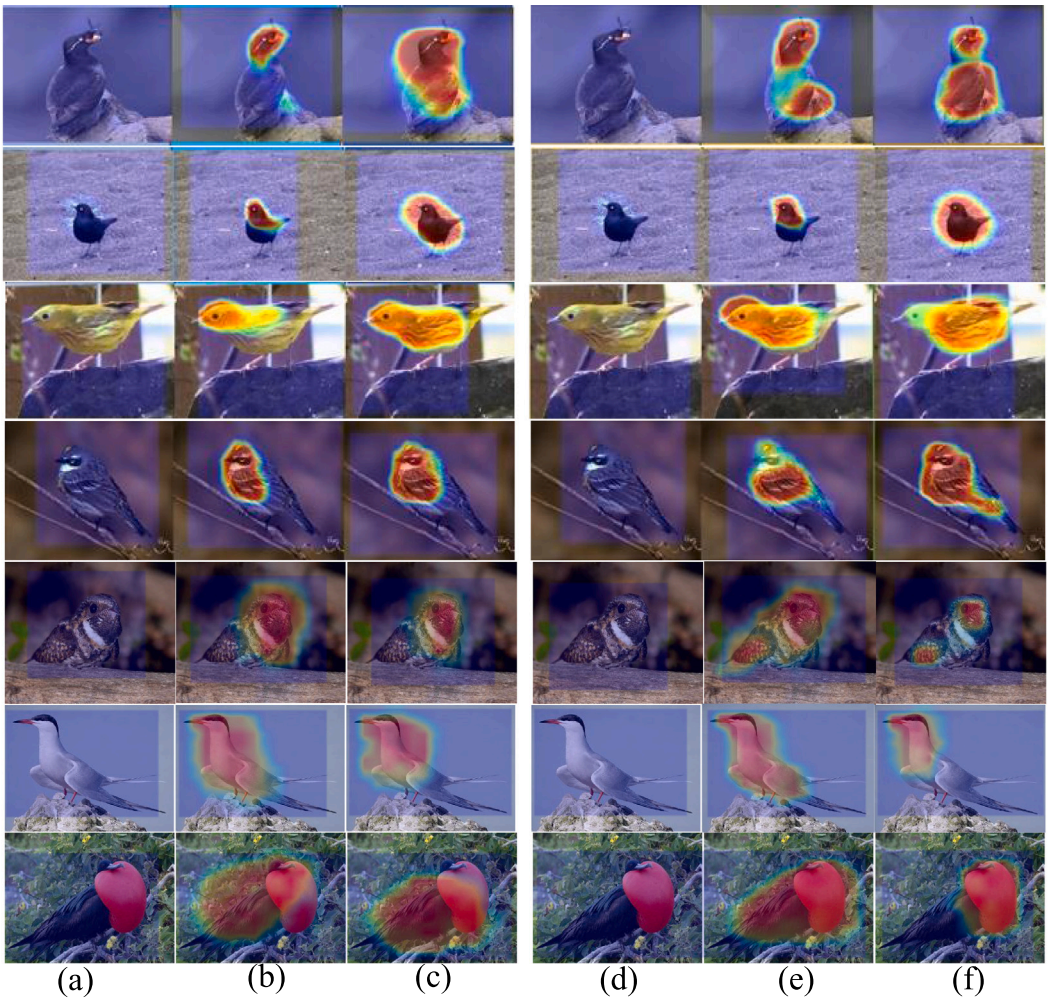



**Fig. 8.** Grad-CAM results. (a) and (d) are the original images, (b) and (c) are the Grad-CAM of TransFG after 30th and 80th epochs. Similarly, (e) and (f) are the Grad-CAM of TransFG with our progressive training method after 30th and 80th epochs.

**Computation complexity**: In this test, we show the computation complexity when adding our training method in the FGR models. From Table 4 we could clearly see that the proposed progressive training method does not involve high computation complexity.

### 5.4. Ablation experiment

In the proposed collaborated optimization model, the filtered dataset $D$ is obtained through submodular optimization, and then further filtered by self-pace optimization to achieve progressive training. To explore the effectiveness of submodular and self-pace optimization, an ablation experiment will be conducted by removing the effect of certain optimization functions to verify their necessity (detailed setting please see Table 5). Specifically, we give four different settings: **setting 1** with self-pace and submodular optimizations, **setting 2** with self-pace optimization but no submodular optimization, **setting 3** with submodular optimization but no self-pace optimization, and **setting 4** without both. The experimental results are shown in Fig. 7. Although the accuracy of setting 4 increase rapidly in the first few rounds, it has a lower training accuracy and slower convergence speed compared to the other three settings in the subsequent training process. Moreover, setting 1 has higher accuracy and fast convergence rate than settings 2 and 3. In conclusion, submodular optimization and feedback are beneficial for model training, and the combination of the two conditions is more effective than using them separately.

### 5.5. Qualitative analysis

Grad-CAM is an effective visualization tool in the field of deep learning, which can visualize the image blocks that the deep learning model focuses on. Here, we adopt this tool to demonstrate that our method can provide strong support to the FGR method in concentrating on discriminative regions with clear semantic meaning. In Fig. 8, we present the Grad-CAM results of

**Table 4**
The experiment for testing computation complexity.

| Backbone | Collaborated optimization | Flops |
|---|---|---|
| swin | w/o | 30.26 |
| swin | w | 49.21 |
| TransFG | w/o | 53.78 |
| TransFG | w | 64.32 |

**Table 5**
The explanation of ablation setting.

| Ablation setting | | | |
|---|---|---|---|
| Backbone | Self-pace | Submodular | Annotation |
| Yes | Yes | Yes | Setting 1 |
| Yes | Yes | No | Setting 2 |
| Yes | No | Yes | Setting 3 |
| Yes | No | No | Setting 4 |

two methods: the API-NET network without any specific training strategies and the API-NET network with our training methods. We randomly select four bird images for visualization, where the first and fourth columns show the original images, and the second and third columns display the heat maps generated after 30 and 80 rounds of training, respectively.

The first four lines are the samples of easy sub-classes. While the last three lines are the samples of difficult sub-classes. Since there exists a high inter-class variance in easy sub-classes, the Grad-CAM of our method is not obviously different from the original API-NET. Original API-NET has strong capability to discriminate easy sub-classes. In difficult sub-classes, our method shows a significant advantage. This mainly contributes to the performance gain. Specifically, our method uses easy groups to train the network before 30th epochs. The attention of our method can cover all of the saliency sub-regions. This indicates a certain generalization capability. However, some background information may also be involved. Further training the challenging sub-classes, attention focus on a small sub-regions and the invalid background interference can be alleviated. In comparison, the original API-NET could not alleviate the background interference.

## 6. Conclusion

In this paper, we propose to integrate the submodular optimization and the self-pace learning into a maximum–minimum optimization framework, which uses active groups to dynamically regulate self-pace learning. This can progressively guide the training process in challenging scenarios. Extensive experiments have shown that our method can be extended to various fine-grained recognition models and SOTA transformers with prominent accuracy enhancement.

## CRediT authorship contribution statement

**Bin Kang:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Songlin Du:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology. **Dong Liang:** Visualization, Validation, Software. **Fan Wu:** Software, Resources, Investigation, Data curation. **Xin Li:** Validation, Software, Formal analysis, Data curation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under grant (Nos. 62171232, 62272229, 62371253); the China Postdoctoral Science Foundation under Grant 2020M681684; Natural Science Foundation of Jiangsu Province under grant BK20222012.

## References

[1] S. Wang, J. Chang, Z. Wang, H. Li, W. Ouyang, Q. Tian, Content-aware rectified activation for zero-shot fine-grained image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 46 (6) (2024) 4366–4380.
[2] S. Joung, S. Kim, M. Kim, I.-J. Kim, K. Sohn, Learning canonical 3d object representation for fine-grained recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1035–1045.
[3] H. Li, M. Li, Q. Peng, S. Wang, H. Yu, Z. Wang, Correlation-guided semantic consistency network for visible-infrared person re-identification, IEEE Trans. Circuits Syst. Video Technol. (2023).
[4] T. Chen, L. Lin, R. Chen, Y. Wu, X. Luo, Knowledge-embedded representation learning for fine-grained image recognition, 2018, arXiv preprint arXiv:1807.00505.
[5] X. Wang, Y. Chen, W. Zhu, A survey on curriculum learning, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2021) 4555–4576.
[6] R. Du, J. Xie, Z. Ma, D. Chang, Y.-Z. Song, J. Guo, Progressive learning of category-consistent multi-granularity features for fine-grained visual classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (12) (2021) 9521–9535.
[7] K. Song, X.-S. Wei, X. Shu, R.-J. Song, J. Lu, Bi-modal progressive mask attention for fine-grained recognition, IEEE Trans. Image Process. 29 (2020) 7006–7018.
[8] Z. Wu, Q. Chen, Y. Liu, Y. Zhang, C. Zhu, Y. Yu, Progressive multi-stage interactive training in mobile network for fine-grained recognition, 2023, arXiv preprint arXiv:2112.04223.
[9] B. Kang, F. Wu, X. Li, Q. Zhou, Progressive training enabled gine-grained recognition, in: Proceedings of the IEEE International Conference on Image Processing, 2022, pp. 876–880.
[10] H. Liu, J. Li, D. Li, J. See, W. Lin, Learning scale-consistent attention part network for fine-grained image recognition, IEEE Trans. Multimed. 24 (2022) 2902–2913.
[11] Y. Liu, L. Zhou, P. Zhang, X. Bai, L. Gu, X. Yu, J. Zhou, Where to focus: investigating hierarchical attention relationship for fine-grained visual classification, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 57–73.
[12] D. Chang, Y. Tong, R. Du, T. Hospedales, Y.-Z. Sun, Z. Ma, An erudite fine-grained visual classification model, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.
[13] T. Yan, H. Li, B. Sun, Z. Wang, Z. Luo, Discriminative feature mining and enhancement network for low-resolution fine-grained image recognition, IEEE Trans. Circuits Syst. Video Technol. 32 (8) (2022) 5319–5330.
[14] A. Bera, Z. Wharton, Y. Liu, N. Bessis, A. Behera, SR-GNN: Spatial relation-aware graph neural network for fine-grained image categorization, IEEE Trans. Image Process. 31 (2022) 6017–6031.
[15] A. Behera, Z. Wharton, P.R. Hewage, A. Bera, Context-aware attentional pooling (cap) for fine-grained visual classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 929–937.
[16] S. Wang, Z. Wang, H. Li, J. Chang, W. Ouyang, Q. Tian, Accurate fine-grained object recognition with structure-driven relation graph networks, Int. J. Comput. Vis. 132 (1) (2024) 137–160.
[17] Y. Zhao, J. Li, X. Chen, Y. Tian, Part-guided relational transformers for fine-grained visual recognition, IEEE Trans. Image Process. 30 (2021) 9470–9481.
[18] R. Ji, L. Wen, L. Zhang, D. Du, Y. Wu, C. Zhao, X. Liu, F. Huang, Attention convolutional binary neural tree for fine-grained visual categorization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10468–10477.
[19] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, J. Jiao, Selective sparse sampling for fine-grained image recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6599–6608.
[20] Y. Zhao, K. Yan, F. Huang, J. Li, Graph-based high-order relation discovery for fine-grained recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15079–15088.
[21] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, TransFG: A transformer architecture for fine-grained recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 852–860.
[22] T. Gong, K. Chen, L. Zhang, J. Wang, Debiased contrastive curriculum learning for progressive generalizable person re-identification, IEEE Trans. Circuits Syst. Video Technol. (2023) early accepted.
[23] L. Anzalone, P. Barra, S. Barra, A. Castiglione, M. Nappi, An end-to-end curriculum learning approach for autonomous driving scenarios, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 19817–19826.

[24] G. Hacohen, D. Weinshall, On the power of curriculum learning in training deep networks, in: Proceedings of the International Conference on Machine Learning, PMLR, 2019, pp. 2535–2544.
[25] T. Matiisen, A. Oliver, T. Cohen, J. Schulman, Teacher–student curriculum learning, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2019) 3732–3740.
[26] A. Tong, C. Tang, W. Wang, Semi-supervised action recognition from temporal augmentation using curriculum learning, IEEE Trans. Circuits Syst. Video Technol. 33 (3) (2023) 1305–1319.
[27] C. Qian, J.-C. Shi, K. Tang, Z.-H. Zhou, Constrained monotone $k$-submodular function maximization using multiobjective evolutionary algorithms with theoretical guarantee, IEEE Trans. Evol. Comput. 22 (4) (2017) 595–608.
[28] Y. Yoshida, Maximizing a monotone submodular function with a bounded curvature under a knapsack constraint, SIAM J. Discrete Math. 33 (3) (2019) 1452–1471.
[29] K.K. Sarpatwar, B. Schieber, H. Shachnai, Constrained submodular maximization via greedy local search, Oper. Res. Lett. 47 (1) (2019) 1–6.
[30] A. Badanidiyuru, J. Vondrák, Fast algorithms for maximizing submodular functions, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 1497–1514.
[31] N. Buchbinder, M. Feldman, J. Naor, R. Schwartz, Submodular maximization with cardinality constraints, in: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 1433–1452.
[32] X. Dong, L. Zheng, F. Ma, Y. Yang, D. Meng, Few-example object detection with model communication, IEEE Trans. Pattern Anal. Mach. Intell. 41 (7) (2019) 1641–1654.
[33] C. Chekuri, K. Quanrud, Parallelizing greedy for submodular set function maximization in matroids and beyond, in: Proceedings of the ACM SIGACT Symposium on Theory of Computing, 2019, pp. 78–89.
[34] J. Krause, M. Stark, J. Deng, et al., 3D object representations for fine-grained categorization, in: ICCV Workshop, 2013.
[35] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The inaturalist species classification and detection dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8769–8778.
[36] C. Chekuri, K. Quanrud, Parallelizing greedy for submodular set function maximization in matroids and beyond, in: Proceedings of the ACM SIGACT Symposium on Theory of Computing, 2019, pp. 78–89.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[38] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[39] W. Ge, X. Lin, Y. Yu, Weakly supervised complementary parts models for fine-grained image classification from the bottom up, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3034–3043.
[40] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, J. Jiao, Selective sparse sampling for fine-grained image recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6599–608.
[41] W. Luo, X. Yang, X. Mo, Y. Lu, L.S. Davis, J. Li, J. Yang, S.-N. Lim, Cross-x learning for fine-grained visual categorization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8242–8251.
[42] Y. Chen, Y. Bai, W. Zhang, T. Mei, Destruction and construction learning for fine-grained image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5157–5166.
[43] R. Du, D. Chang, A.K. Bhunia, J. Xie, Z. Ma, Y.-Z. Song, J. Guo, Fine-grained visual classification via progressive multi-granularity training of jigsaw patches, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 153–168.
[44] S. Huang, X. Wang, D. Tao, Snapmix: Semantically proportional mixing for augmenting fine-grained data, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 2, 2021, pp. 1628–1636.
[45] W. Luo, H. Zhang, J. Li, X.-S. Wei, Learning semantically enhanced feature for fine-grained image classification, IEEE Signal Process. Lett. 27 (2020) 1545–1549.
[46] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.
[47] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, Transfg: A transformer architecture for fine-grained recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 1, 2022, pp. 852–860.
[48] H. Sun, X. He, Y. Peng, Sim-trans: Structure information modeling transformer for fine-grained visual categorization, in: Proceedings of the ACM International Conference on Multimedia, 2022, pp. 5853–5861.
[49] Q. Xu, J. Wang, B. Jiang, B. Luo, Fine-grained visual classification via internal ensemble learning transformer, IEEE Trans. Multimed. 25 (2023) 9015–9028.
[50] S. Kim, J. Nam, B.C. Ko, Vit-net: Interpretable vision transformers with neural tree decoder, in: International Conference on Machine Learning, PMLR, 2022, pp. 11162–11172.

**Bin Kang** received the M.S. degree in Circuits and Systems, and the Ph.D. degree in Electrical Engineering from Lanzhou University and Nanjing University of Posts and Telecommunications, in 2011 and 2016, respectively. He is currently an associate professor at the College of Internet of Things, Nanjing University of Posts and Telecommunications. His research interests include computer vision and pattern recognition.

**Sonlin Du** received the Ph.D. degree in physics from Lanzhou University, Lanzhou, China, and the Ph.D. degree in engineering from Waseda University, Tokyo, Japan, in 2019. He is currently with the School of Automation, Southeast University, Nanjing, China. He was the recipient of the Best Paper Award at ISPACS2017, the Best Presentation Award at ICISIP2018, the Excellent Paper Award at ISIPS2018, and the Best Presentation Award at the Computer Vision Session of AIVR2019. His research focuses on developing computer vision algorithms and high-speed vision systems.

**Dong Liang** received the B.S. degree in Telecommunication Engineering and the M.S. degree in Circuits and Systems from Lanzhou University, China, in 2008 and 2011, respectively. He is studying for the Ph.D. course at the System Sensing and Control Lab, Graduate School of Information Science and Technology, Hokkaido University, Japan. His research interests include computer vision and pattern recognition. Dong Liang received the B.S. degree in telecommunication engineering and the M.S. degree in circuits and systems from Lanzhou University, Lanzhou, China, in 2008 and 2011, respectively, and the Ph.D. degree from the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan, in 2015. He is currently an associate professor with the Pattern Recognition and Neural Computing Laboratory, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His research interests include computer vision and pattern recognition

**Fan Wu** is studying for the M.S. course at the College of Internet of Things, Nanjing University of Posts and Telecommunications, China. Her research interests include computer vision and pattern recognition.

**Xin Li** received the M.S. and Ph.D. degree in Computer Science from Nanjing University of Science and Technology. He is currently an associate professor at the College of Internet of Things, Nanjing University of Posts and Telecommunications. His research interests include data analysis and pattern recognition.