Graph-based High-Order Relation Discovery for Fine-grained Recognition

Yifan Zhao[1] Ke Yan[2] Feiyue Huang[2] Jia Li[1,3*]
[1]State Key Laboratory of Virtual Reality Technology and Systems, SCSE, Beihang University
[2]Tencent Youtu Lab, Shanghai, China
[3]Peng Cheng Laboratory, Shenzhen, China
{zhaoyf, jiali}@buaa.edu.cn, {kerwinyan, garyhuang}@tencent.com

Abstract

Fine-grained object recognition aims to learn effective features that can identify the subtle differences between visually similar objects. Most of the existing works tend to amplify discriminative part regions with attention mechanisms. Besides its unstable performance under complex backgrounds, the intrinsic interrelationship between different semantic features is less explored. Toward this end, we propose an effective graph-based relation discovery approach to build a contextual understanding of high-order relationships. In our approach, a high-dimensional feature bank is first formed and jointly regularized with semantic- and positional-aware high-order constraints, endowing rich attributes to feature representations. Second, to overcome the high-dimension curse, we propose a graph-based semantic grouping strategy to embed this high-order tensor bank into a low-dimensional space. Meanwhile, a group-wise learning strategy is proposed to regularize the features focusing on the cluster embedding center. With the collaborative learning of three modules, our module is able to grasp the stronger contextual details of fine-grained objects. Experimental evidence demonstrates our approach achieves new state-of-the-art on 4 widely-used fine-grained object recognition benchmarks.

1. Introduction

Fine-grained object recognition focuses on distinguishing and classifying objects of a basic-level category into subclasses, which is a challenging task due to the subtle visual differences among different classes. Benefiting from the strong perceptual capability of deep neural networks, handling subtle variances using visual features [42, 34, 49, 19] has made significant progress. In particular we consider two popular families of methods in tackling this problem, *i.e.*, discriminative part learning and feature representation learning.

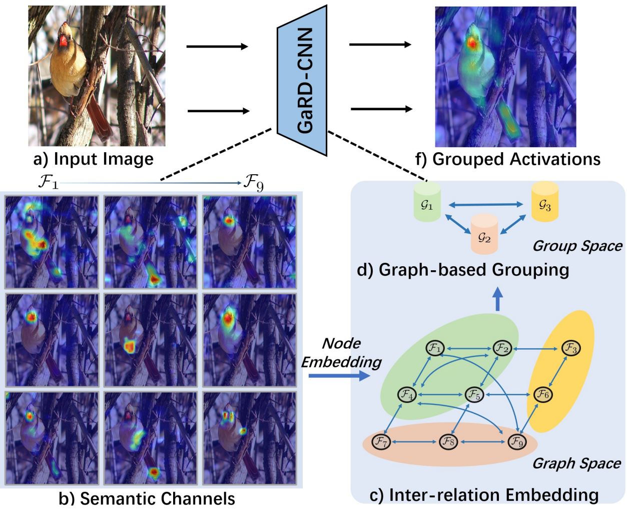


Figure 1. The motivation of proposed approach. Our proposed approach first exploits the structurally channel-aware relationship b) into a high-dimensional graph embedding. Then these relation nodes are grouped into low-dimensional space d) with a semantic grouping strategy, forming the final grouped activations f).

Studies of the first family [48, 42, 34, 6] usually deal with the fine-grained categorization problem by localizing distinct parts. Some representative methods tend to utilize the part detectors [13, 18] or segmentation parsers [19, 21] in different categorization tasks. With accurate part parsing results obtained, satisfactory performance for fine-grained classifiers could also be achieved simultaneously. Besides methods using manual annotations, attention-based approaches [19, 42, 34, 6, 31] show its ability in discovering object parts during weakly-supervised training. However, part localizations using attention mechanisms perform unreliable results under complex scenarios. As networks fail to capture the correct part localizations, further strengthening these regions would lead to catastrophic overfitting.

[*] Correspondence should be addressed to Jia Li. URL: http://cvteam.net

The other family of approaches [28, 46, 36, 49] usually tackle the classification problem as a representation learning task. As object parts emerge naturally in different feature channels [14] during the weakly-supervised classification task, exploiting mutual relationships among different channels [36, 12, 3, 50] is meaningful and beneficial for fine-grained feature representations. As one of the predominant methods, bilinear pooling [28] exploits the second-order classification features from two different networks. Moreover, Zheng *et al.* [49] propose a trilinear attention mechanism, using third-order pooling to build channel relationships. However, high order features would lead to high dimensions (*e.g.*, $C \times C$ dimensions for homogeneous features $\mathbf{X} \cdot \mathbf{X}^\top, \mathbf{X} \in \mathbb{R}^{C \times W \times H}$), bringing in heavy computation burden and overfitting risks. Thus two major concerns arise: 1) how to build global relation scopes using high-order relationships and 2) how to embed the high-dimension features in a low-dimension manifold?

In this paper, we propose a graph-based relation discovery (GaRD) approach to excavating finer relational attributes from intrinsic network features. As illustrated in Fig. 1, different from previous research, our approach tackles the representative feature learning in an *expansion and compression* manner. Inspired by the emerging semantic parts in class activation maps (Fig. 1 b)), our *expansion* motif is to construct contextual relationships among multiple feature channels by learning high-order representations. However, the tentative nature of channel-aware mechanisms [28, 49] tends to omit the spatially structural information and use averaged logits to represent each channel. To overcome this natural defect, we propose a relation-discovery module where the structural relations are constructed by employing a position-aware gating operation, providing high-order spatial enhancement for further channel interactions. Meanwhile, heterogeneous features from different levels are adopted to build a cross-channel relation with positional enhanced features. Finally, a mix-order tensor bank $\mathbf{T} \in \mathbb{R}^{C_1 \times C_2}$ is formed, endowing rich features but resulting in redundant high dimensions.

To address this problem, existing classification models *compress* high-dimension features with feature factorizations [27] or low-rank representations [23]. Despite their performance deficiencies, semantic relationships among different features are less taken into consideration, which is crucial in fine-grained vision tasks. To explore the semantic relationships, we first formulate the mix-order tensor into a graph representation in Fig. 1 c) and then propose a graph grouping module to adaptively embed the high-order relation matrices into a low-dimension manifold. The graph convolutional layer efficiently encodes this relation matrix with a densely-connected relational graph. To redivide these nodes into different groups, we adopt an auxiliary graph layer to learn the grouping rules based on their semantic similarities. Hence the mix-order feature bank is embedded in a low-dimension manifold while retaining its rich semantic relationships for fine-grained recognition. Beyond these two modules, we first advocate employing the group-wise training mechanism for fine-grained image classification without additional regularizations, which utilize the center of grouped images instead of per image samples for gradient descent. This mechanism alleviates overfitting and gradient anomalies caused by hard samples. Experimental evidence demonstrates the proposed approach achieves state-of-the-art results on four popular benchmark datasets, *i.e.*, CUB-200-2011 [38], Stanford-Cars [25], Aircrafts [30], and NAbirds [37].

In summary, our contribution is threefold: 1) We propose a novel graph-based relation discovery (GaRD) approach for fine-grained recognition, which adaptively exploits the relation-aware feature embeddings to enhance the discriminative representation abilities. 2) We propose to learn the positional and semantic feature relationships with an effective relation-discovery module, and learn a semantic grouping rule to cluster the high-order relationships. 3) We propose a simple yet effective group-wise learning strategy to update gradient using cluster center prototypes, alleviates overfitting and anomalies caused by hard samples.

2. Related Work

Discriminative part learning. Deep CNN has its natural ability in localization discriminative in the wake of the learning process of classification. Visual parts emerge naturally [14] during the gradient backward process. In turn, accurate localizing these discriminative parts can be of great help to the recognition task. Previous pioneer works [17, 47, 18, 41, 13, 15] tend to utilize the bounding box information by weak supervisions and manual annotations. For example, Zhang *et al.* [47] propose part-based R-CNNs to localize and detect the whole object and related parts, enforcing learned geometric constraints for accurate representation. Lam *et al.* [26] proposed a sequential search for informative parts with bounding box annotations and embedded the heuristic function into the LSTM network. Recent part-based methods [19, 42, 34, 6, 31] usually handle these tasks using attention mechanisms to visualize the class activation maps. For example, a dual-stage attention framework [42] is conducted to filter part patches that are most relevant to the semantic object. Simon *et al.* [34] propose an unsupervised part model discovery method for deep neural activation maps. In this method, deep neural activation maps are used to exploit the channels of classification networks as a part detector. Fu *et al.* [10] propose a hierarchical structure to automatically locate the most useful part by adopting an attention proposal sub-network.

Feature representation learning. Leading by bilinear pooling techniques [28], feature representation meth-

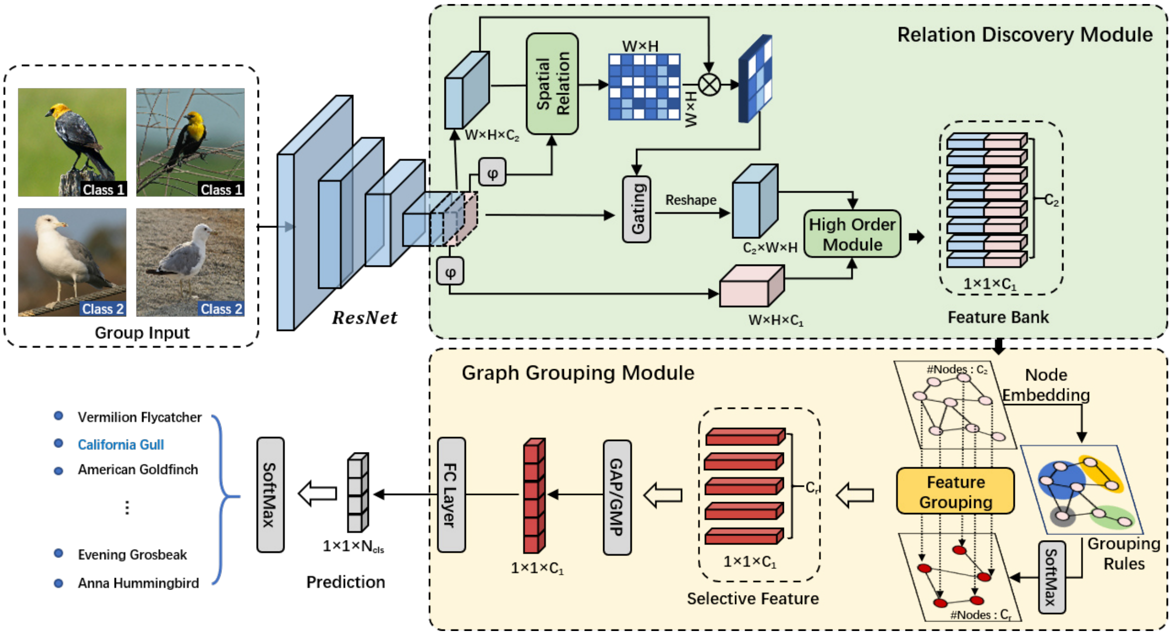


Figure 2. The proposed graph-based relation discovery (GaRD) approach consists of three essential modules: the relation-discovery module to extract rich relation-aware high-dimension features, the graph-based semantic grouping module to find low-dimension feature embeddings, and the group-wise learning strategy is adopted to update the gradient using class centers.

ods [46, 36, 49, 45] in high-order shows stronger generalization and categorization abilities. For example, compact bilinear features [11] are proposed to reduce the feature dimensions by compacting two homogeneous tensor sketches. Kong *et al.* [23] proposed a low-rank representation method, capturing second-order statistics with Frobenius norm projection. Besides these efforts, Zheng *et al.* [49] proposed to adopt the trilinear attention with the third-order pooling to construct the channel attention responses. There are some other methods [3, 40, 36, 20, 4, 29] focusing on the multi-scale or multi-channel representations. OSME [36] module applies an effective multi-attention multi-class constraint to regularize the feature learning. Chang *et al.* [3] propose a mutual interaction mechanism to exploit the feature relation across different channels. Beyond the above methods, exploiting different pooling techniques [46] and building feature regularizations [8, 9] also greatly enhance the final feature representations. Different from these aforementioned researches, in this paper, we propose to discover the high-order relationship while learning a semantic grouping for discriminative feature representations.

3. Approach

In this section, we introduce a graph-based relation discovery (GaRD) approach for fine-grained recognition in Fig. 2. The first key idea of our approach is to exploit inter-relations among different semantic and structural features in Section 3.1, depicting this relation with high-order rich features. As these features are catastrophic high-dimensional which are usually hard to optimize, we present a new graph-based semantic grouping module to embed these features in a compact space in Section 3.2. Beyond these improvements, in Section 3.3, we propose a group-wise learning strategy to alleviate the outliers in the gradient descent optimization.

3.1. Relation Discovery

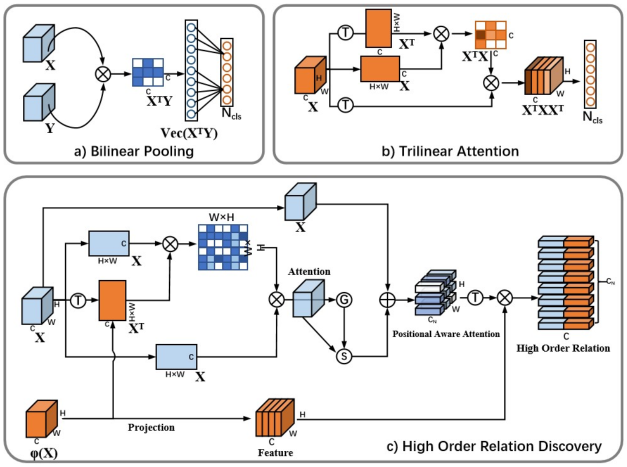
Given an input image $\mathcal{I}$, let $\mathbf{X} \in \mathbb{R}^{W\times H\times C}$ be the C-dimensional with $H \times W$ feature planes encoded by a backbone network $\mathbf{X} = \Phi(\mathcal{I})$. Thus the most common way for classification is to embed the final feature $\mathbf{X}$ by using global pooling operations (GAP or GMP), calculating mean or maximum values on the $H \times W$ feature plane.

High-order attentions. Adopting mean or maximum pooling operations usually fails to capitalize on the interactions among different semantic channels. To exploit the semantic response among channels in Fig. 3 a), second-order matrices $\mathbf{F}^r$ for each location (i, j) builds an inter-channel relationship by transposing itself $\Phi_A(\cdot) \in \mathbb{R}^{WH\times C_A}$ and multiplying with another CNN stream $\Phi_B(\cdot) \in \mathbb{R}^{WH\times C_B}$. The final features then pass a fully-connected (`fc`) layer for the final N_{cls}-way classification:

$$\mathbf{F}^r = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\texttt{vec}(\Phi_A(\mathcal{I})_{i,j}^{\top}\Phi_B(\mathcal{I})_{i,j}), \qquad (1)$$

$$\mathbf{F}^b = \mathbf{W}\cdot\mathbf{F}^r + \mathbf{b},$$

where $\texttt{vec} : \mathbb{R}^{C_A\times C_B} \rightarrow \mathbb{R}^{C_A C_B\times 1}$ denotes the vector-


a) Bilinear Pooling

b) Trilinear Attention

c) High Order Relation Discovery

Figure 3. Illustrations of different mutual attention methods. a) Bilinear pooling [28]: building channel-aware second-order relations, using vectorized features. b) Trilinear attention [49]: third-order channel relations, preserving the original feature shape. c) Our relation-discovery module: joint positional and channel-relation aware, forming a relational Tensor bank.

ization of second-order matrices. $\mathbf{W} \in \mathbb{R}^{C_A C_B \times N_{cls}}$ is the learnable weight of the fc layer. Although rich features are obtained, learning such high-dimension features can easily lead to inferior optimization. Inspired by the non-local operations, trilinear attention-based methods [49, 12] regard the cross channel relationships as an attention map $\mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{C\times C}$, generated from the same feature map (see Fig. 3 b)). The channel-aware attention map then attaches different importance to the original feature $\mathbf{X}$, resulting in the third-order results $\mathcal{S}(\mathbf{X}^\top\mathbf{X})\mathbf{X}^\top \in \mathbb{R}^{WH\times C}$, where $\mathcal{S}$ denotes the softmax normalization.

Joint relation discovery. The first drawback of cross-channel relations lies in the omitting of positional information. As in Eqn. (1), each pixel is treated equally with an averaged summation over $W \times H$. However, as object parts emerge automatically in network features, encoding original features with positional importance is thus necessary. The positional attention weights can be represented as:

$$\mathbf{P} = \mathcal{N}(\mathcal{M}(\frac{1}{C}\sum_{\mathbf{i}=1}^{C}\mathbf{X}_i^\top\varphi(\mathbf{X}_i))) \in \mathbb{R}^{WH\times WH}, \quad (2)$$

where $\mathcal{M}(x) = \mathtt{sign}(x)x^{-1/2}$ and $\mathcal{N}(x) = x/||x||_2^2$ are the moment and L2 normalization respectively. In addition, different network layers present object semantics in different scales, where the latter one has a larger receptive field. Leveraging cross-layer semantics also enhances the representation of multi-scale learning. Here we use $\varphi(\mathbf{X})$ denote the latter layer than $\mathbf{X}$ from the same ResNet stage. Thus $\mathbf{P}$ serves as an attention weight to find the spatial correlations across different layers. The positional weights then strengthen the original feature with a spatial attention and then pass an adaptive gating operation to select (symbol S in Fig. 3) the most useful features when occurring different samples:

$$\mathbf{E}_i = \mathcal{G}(\sum_{c=1}^{C}\mathbf{P}_c)\cdot(\mathbf{P}_i\mathbf{X}_i^\top) + \mathbf{X}_i, \quad (3)$$

where $\mathcal{G}(\mathbf{P}) \in \mathbb{R}^1$ denotes the gating weight generated by a $\mathtt{fc}$ layer. The gating operation is generated based on the spatial perceptions to form the positional aware features.

The second drawback is that the cross-channel interactions in trilinear attentions are described implicitly, using a re-weighting mechanism for each channel. This attention mechanism can be regarded as denoising or high-pass filtering operations. Although features are maintained in original shape $\mathbb{R}^{W\times H\times C}$, the relationship matrix across different semantic channels is omitted. Thus we propose to use the rich relation-aware representation instead of the common feature map, using an explicit tensor bank $\mathcal{T}$ for relation description. After obtaining the positional enhanced feature $\mathbf{E}$, the relation matrix can be built by similar operations:

$$\mathcal{T} = \mathcal{N}(\mathcal{M}(\frac{1}{WH}\sum_{\mathbf{i}=1}^{WH}\mathbf{E}^\top\varphi(\mathbf{X}_i))) \in \mathbb{R}^{C_N\times C}, \quad (4)$$

where C_N denotes the channel dimension of positional aware attention. Unlike conventional bilinear pooling methods that perform vectorization or matrix factorization, we construct a tensor bank $\mathcal{T}$ with C_N tensors. Each tensor has the same C-dimensions for semantic mappings which are corresponded to the original feature channels.

3.2. Graph-based Tensor Grouping

The most common method to embed a high-dimension feature $\mathcal{T} \in \mathbb{R}^{C_N\times C}$ is to employ MLPs with fc layers. However, as mentioned in Eqn. (1), this embedding $\mathbf{W}$ would introduce enormous learnable parameters $C^2 \times N_{cls}$, which are usually hard to optimize for fine-grained classification tasks with limited data. Thus a natural concern arises, how to embed the tensor bank into a low-dimension embedding and keep its semantic mapping as well?

When the tensor bank $\mathcal{T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{C_N}\}$ is constructed, it can be formulated as a graph with C_N nodes of C-dim. As it stands, it is the dimension of attention parts C_N that leads to the optimization complexity. Notably, it can be found that these nodes essentially share much mutual information, *e.g.*, responding to the same object part in CAM [33]. We thus propose to aggregate these features based on their similarity using Kipf's *et al.* [22] Graph Convolutional Networks. The pair-wise adjacent relationship between different nodes can be defined as:

$$\mathbf{A}_{i,j} = \frac{\tau(\mathbf{x}_i)^\top \cdot \tau(\mathbf{x}_j)}{\|\tau(\mathbf{x}_i)\| \, \|\tau(\mathbf{x}_j)\|}, \tag{5}$$

where $\tau(\cdot)$ denotes a 1×1 convolution layer for dimensional transformation. The final adjacent matrix can be defined by adding self-loop as $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\mathbf{I} \in \mathbb{R}^{C_N \times C_N}$ is an identity matrix. With this dense-connected GCN operation, each node can be updated by this similarity-based aggregation:

$$\mathbf{H} = \texttt{ReLU}(\widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{T} \mathbf{W}^g). \tag{6}$$

$\mathbf{W}^g \in \mathbb{R}^{C \times d_h}$ is the learnable graph weights with the hidden dimension d_h, and $\widetilde{\mathbf{D}} = \sum_j \widetilde{\mathbf{A}}_{\mathbf{i},\mathbf{j}}$ is the diagonal matrix for normalization. $\mathbf{T}$ denotes the matrix form of the tensor bank $\mathcal{T}$. Thus the feature of each node is updated by this message passing operation. The other aim of graph embedding operation [2, 44] is to form multiple groups to get the contextual understanding. We further propose to learn these grouping rules by learning a new graph convolutional layer, which is desired to find an embedding $\mathbb{R}^{C_N \times d_h} \to \mathbb{R}^{C_r \times d_h}$. Hence we use the aggregated features with its adjacent matrix to form this embedding $\mathbf{G}$:

$$\mathbf{G} = \texttt{ReLU}(\widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H} \mathbf{W}^{emb}). \tag{7}$$

$\mathbf{W}^{emb} \in \mathbb{R}^{d_h \times C_r}$ is the graph weights. Thus $\mathbf{G} \in \mathbb{R}^{C_N \times C_r}$ defines a mapping function of each node from original feature space to form the new graphlets:

$$\mathbf{Z} = \mathbf{H}^\top \frac{e^{\mathbf{G}_{i,j}}}{\sum_{j=1}^{C_r} e^{\mathbf{G}_{i,j}}} \in \mathbb{R}^{d_h \times C_r}, \tag{8}$$

where C_r is the number of new embedded graphlets. *i.e.*the original high-dimension relational matrices are clustered into C_r semantic groups, while C_r is set as $\lfloor C_N/r \rfloor$ empirically. We conduct a softmax operation in the group dimension, indicating that each new group is composed of a probabilistic combination of original C_N Nodes. In this manner, the high-order tensor bank $\mathcal{T}$ can be easily assigned in a low-dimension manifold, and mapped by both channel dimension and node dimension. The grouped feature $\mathbf{Z}$ is then performed with residual connections to construct the final embedding $\widetilde{\mathbf{Z}} = \mathbf{Z} + \mathbf{H}$. Similar to conventional classification tasks, thus these embedding can be measured by passing the final pooling layer, *i.e.*, GMP and GAP, and a classifier to predict the probability of N_{cls} classes.

3.3. Group-wise Learning

Few research efforts have paid their attention to the exploitation on pair-wise relationships [36, 35, 32] or introducing pair-wise confusions [8] in fine-grained recognition tasks. As shown in Fig. 4 a), methods of the first group [36, 35] proposed to constrain the intra-class similarity and inter-class dissimilarities, which samples an anchor image (denotes as $\mathcal{A}$) to find negative and positive pairs. In Fig. 4 b), pair-wise relations [8] are proposed to regularize features from different classes $\mathcal{F}_1$ and $\mathcal{F}_2$ in similar distributions. Despite their high-computation costs in sampling pairs, the group-wise correlations are less explored, which sometimes results in a bad feature embedding due to the restriction of data distributions.

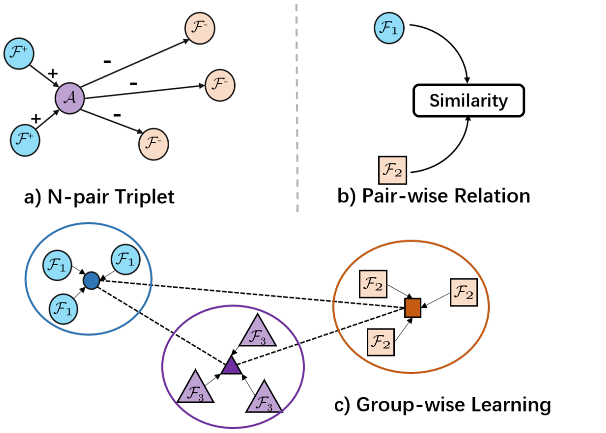


Figure 4. Illustrations of three typical pair-wise regularizations: a) Pair-wise triplet regularization [35, 36], b) Pair-wise relations [8], and c) proposed group-wise learning.

To revisit the fine-grained recognition task from a new perspective, here we propose a group-wise learning strategy in Fig. 4 c). One clear problem in fine-grained classification is the overfitting of hard cases. A preferable class cluster should be centralized by representative samples while omits the outlying hard samples. Unlike previous works using pair-wise constraints, we propose to use the group-wise training in each mini-batch during gradient descent. It means that we use the center of multiple samples of the specific class as a mean feature for updating the network parameters. This operation can be naturally embedded in the network with the cross-entropy loss function $\mathcal{L}_{\texttt{CE}}$. A typical group-wise learning loss in a minibatch is first random select N classes and then sample K images in each class, thus the loss function can be represented as:

$$\mathcal{L}_{\texttt{Batch}} = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathcal{L}_{\texttt{CE}}(\mathcal{F}(\mathbf{y}_n | \mathcal{I}_{n,k}), \mathbf{y}_n), \tag{9}$$

where $\mathcal{F}$ and $\mathbf{y}$ denote the embedded features and labels. With this group-wise training mechanism, networks tend to form clustered embeddings of each class rather than discrete instance-level ones.

4. Experiments

4.1. Experimental settings

Dataset. In this paper, we conduct experiments on four public popular benchmarks: 1) CUB-200-2011 [38] contains 11,788 images from 200 wild bird species, which is the widely-used benchmark for its representativeness, 2) Stanford-Cars [25] contains 16,185 images of 196 car subcategories, 3) FGVC-Aircraft [30] contains 10,000 images of 100 aircraft classes and 4) NA-birds [37] is a large dataset with 48,562 images for over 555 bird classes. We follow the standard dataset partition as in the original works.

Implementation details. We adopt ResNet-50 [16] network pretrained on ImageNet [5] as our backbone for all experiments. We use the SGD optimizer with the initial learning rate of $8e-4$ annealed by 0.1 every 60 epochs (overall 240 epochs) and momentum is set as 0.9. The training and testing protocol follow the state-of-the-art works [29, 4, 50] using random cropping of 448×448 in training and center crop during inference. We adopt the commonly used techniques, *i.e.*random cropping and erasing, left-right flipping, color jittering for data augmentation. For fair comparisons, we report results of ResNet-50 with identical training and data augmentation protocols as our baseline. Our model is trained end-to-end without any part or bounding box annotations on 2 NVIDIA TITAN Xp GPUs for acceleration. We set $N = 4$, $K = 4$ with a batchsize of 16 in the group-wise training for the first three datasets and set N as 8 for the larger NA-birds dataset. We select 10% of the training set as validation for fine-tuning the hyperparameters.

4.2. Comparison with State-of-the-art

CUB-200-2011 dataset. Here we roughly divide the model into two typical types, *i.e.*methods using part localization cues and feature-based representation learning. The comparison results with 16 state-of-the-art methods are exhibited in Tab. 1. It is noted that multi-crop enhancement is adopted in previous works [10, 48, 43] to boost performance. It can be seen that the part-based methods achieve comparable results with the feature-based model, which indicates that learning a preferable feature embedding is the key problem in fine-grained classification.

FGVC-Aircraft dataset. Tab. 2 reports the results on FGVC-Aircraft dataset. Similar to the performance on CUB, recent feature-based learning methods DCL [4] and Cross-X [29] achieves the accuracy of 93.0% and 92.7%, which is much higher than the previous work [28] of 84.1%. Note that many of the performance gains in recent models may come from different training schemes or backbone networks. We report the ResNet-50 baseline in Tab. 5 with an accuracy of 90.7%, which is higher than the earlier works. Starting from this high-baseline, our final model achieves the performance of 94.3%, which is a clear improvement compared to state-of-the-art works [4, 29].

Table 1. Performance on CUB-200-2011 dataset. 1-Stage: one-stage end-to-end training methods. †: using additional annotation. ‡: introducing multiple backbone layers.

Type	Method	1-stage	Accuracy
Part Based	PA-CNN† [24]		84.3%
	RA-CNN [10]		85.3%
	MA-CNN [48]	✓	86.5%
	Interpret [19]		87.3%
	NTSNet [43]		87.5%
	DF-GMM [40]		88.8%
	S-LSTM‡ [13]		90.4%
Feature Based	Bilinear [28]	✓	84.0%
	MAMC [36]	✓	86.5%
	MaxEnt-Dense161 [9]	✓	86.5%
	PC-Dense161 [8]	✓	86.9%
	HBP [45]	✓	87.1%
	DFL-CNN [39]	✓	87.4%
	Cross-X [29]	✓	87.7%
	DCL [4]	✓	87.8%
	TASN [49]		87.9%
	ACNet [20]	✓	88.1%
	S3N [6]	✓	88.5%
	Ours (ResNet50)	✓	89.6%

Table 2. Performance on FGVC-Aircraft dataset. 1-Stage: one-stage end-to-end training methods.

Type	Method	1-stage	Accuracy
Part Based	RA-CNN [10]		88.2%
	MA-CNN [48]	✓	89.9%
	NTSNet [43]		91.4%
	DF-GMM [40]		93.8%
Feature Based	Bilinear [28]	✓	84.1%
	PC-Dense161 [8]	✓	89.2%
	MaxEnt-Dense161 [9]	✓	89.8%
	DFL-CNN [39]	✓	92.0%
	ACNet [20]	✓	92.4%
	Cross-X [29]	✓	92.7%
	S3N [6]	✓	92.8%
	DCL [4]	✓	93.0%
	API-Net [50]	✓	93.0%
	PMG [7]	✓	93.4%
	Ours (ResNet50)	✓	**94.3%**

Stanford-Cars dataset. Stanford-Cars [25] is a real-world dataset composed of 196 car categories. While the CUB dataset contains more complex scenarios or background confusions. As shown in Tab. 3, it can be easily found that the earlier pioneer works [48, 36, 9] achieve high results over 92.8%, making these methods undifferentiated in recognition abilities. However, our model still shows a clear improvement of state-of-the-art models.

NA-birds dataset. Compared to CUB-200-2011, NA-birds [37] is a relatively larger dataset with over 500 sub-

Table 3. Performance on Stanford-Cars dataset. †: additional bounding box or segmentation annotation. 1-Stage: one-stage end-to-end training methods.

Type	Method	1-stage	Accuracy
Part Based	PA-CNN† [24]		92.8%
	MA-CNN [48]	✓	92.8%
	NTSNet [43]		93.9%
	DF-GMM [40]		94.8%
Feature Based	Bilinear [28]	✓	91.3%
	PC-Dense161 [8]	✓	92.9%
	MaxEnt-Dense161 [9]	✓	93.0%
	MAMC [36]	✓	93.0%
	HBP [45]	✓	93.7%
	TASN [49]		93.7%
	DFL-CNN [39]	✓	93.8%
	Cross-X [29]	✓	94.5%
	DCL [4]	✓	94.5%
	ACNet [20]	✓	94.6%
	S3N [6]	✓	94.7%
	Ours (ResNet50)	✓	**95.1%**

Table 4. Performance on NA-birds dataset. 1-Stage: one-stage end-to-end training methods. †: using additional annotations.

Type	Method	1-stage	Accuracy
Part	PN† [37]		75.0%
	MGE-CNN [46]		**88.0%**
Feature Based	AlexNet-fc6 [37]	✓	35.0%
	Bilinear [28]	✓	80.9%
	Presence† [1]	✓	81.5%
	ResNet-50 [16]	✓	82.2%
	PC-Dense161 [8]	✓	82.8%
	MaxEnt-Dense161 [9]	✓	83.0%
	API-Net [50]	✓	86.2%
	Cross-X [29]	✓	86.2%
	Ours (ResNet50)	✓	**88.0%**

categories. Previous networks usually face difficulties to handle the classification of enormous sub-categories. Note that the performance of ResNet-50 in this table is reported by [29]. Recent feature-based methods [50, 29] shows its generalization ability in handling this task, reaching 86.2%. Comparing to these methods, our model shows a clear improvement with the top-1 accuracy of 88.0%, verifying the generalization ability of the proposed approach.

4.3. Performance Analysis

Effects of different components. To evaluate the effectiveness of proposed modules, we first employ ResNet-50 with the identical training protocol as baseline model, *e.g.*, 85.4% on CUB dataset. It can be found in Tab. 5 that adopting the bilinear pooling [28] for relation embedding, the performance boost by 2%. While exploiting the relation discovery module can notably improve the baseline performance. Besides, the group-wise training strategies also improve the focusing attention, providing a stable improvement on final results. Based on this high-performance baseline, we further add the graph-based grouping strategy instead of the basic fusion operations, which provides a steady improvement.

Table 5. Ablation studies of our different components on three benchmarks. $\mathcal{M}_{Rel}$, $\mathcal{M}_{Graph}$, $\mathcal{M}_{Group}$ denotes the proposed relation-discovery module, graph grouping module and group-wise learning respectively.

$\mathcal{M}_{Rel}$	$\mathcal{M}_{Graph}$	$\mathcal{M}_{Group}$	CUB	Aircraft	NAbirds
-	-	-	85.4%	90.7%	83.2%
Bilinear	-	-	87.0%	92.2%	85.5%
✓	-	-	88.1%	92.6%	86.9%
✓	-	✓	88.8%	93.7%	87.4%
✓	✓	✓	89.6%	94.3%	88.0%

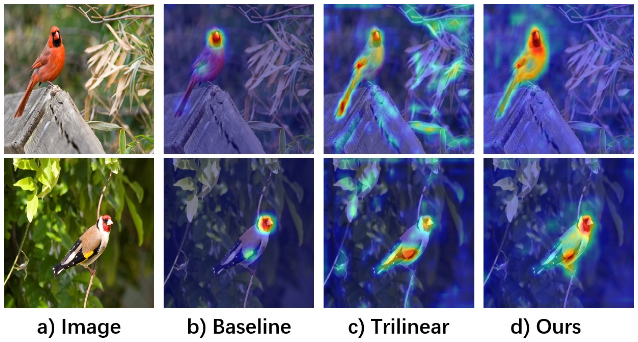


Figure 6. Illustration of three feature embedding strategies: a) baseline, b) trilinear attention [49], and c) Our model.

Effects of feature embedding. The most crucial issue in feature-based learning approaches is to find appropriate feature embeddings. We explore different kinds of settings in Tab. 6. Starting from the baseline, we re-implement the trilinear attention [49] on our high baseline settings, which improves the attention regions and forms a global scope for object recognition (Fig. 6).

Another main exploration is to find embedding ways of our relation module, it is interesting that using MLP with `fc` layers to learn this high embedding would lead to inferior performance of 88.1% compared to the simple mean aggregation embedding of 88.6%. Moreover, we simply remove the spatial enhancement in constructing high order relationships, resulting in 0.5% lower result. Applying the single-level aggregation without multiple layers will also slightly harm the final performance. On the other hand, we replace the mean aggregation strategy with a static graph embedding (denoted as static). This densely-connected graph is then degraded as learnable `fc` layers and leads to inferior results. While our final model with graph-based message

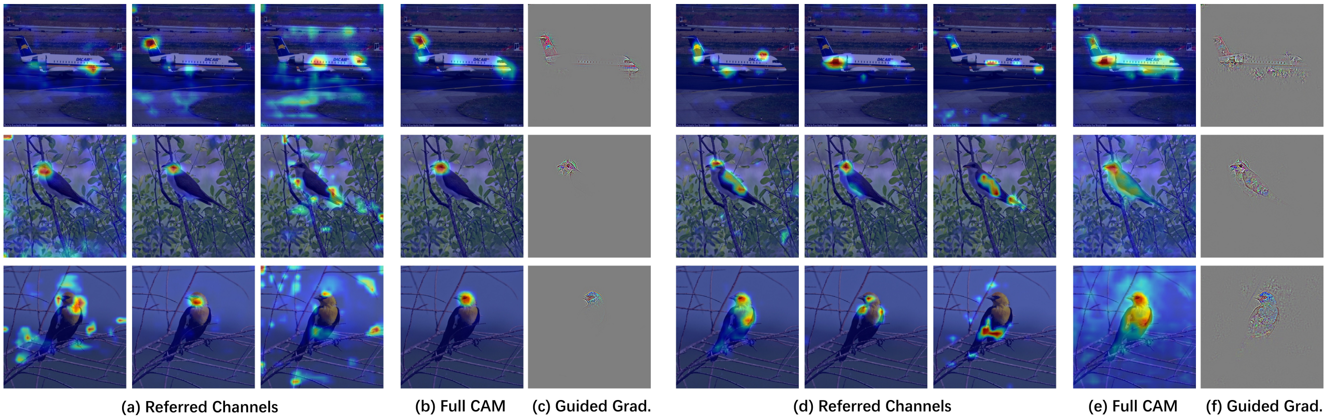


Figure 5. Class activation visualizations of baseline (left) and our model (right). a) and d) are referred top channels generated by baseline and our model. b) and e) are the class activation map of all channels, and c) and f) are the guided gradient of baseline and our model.

Table 6. Performance analysis of feature extraction and feature embedding methods on CUB-200-2011 benchmark. Static: replacing the graph embedding $\mathbf{A}$ with a C_N-by-C_N matrix of ones.

Feature Extraction	Embedding	Accuracy
Baseline (ResNet50)	GAP	85.4%
+$\mathcal{M}_{Group}$	GAP	86.4%
+$\mathcal{M}_{Group}$+Trilinear [49]	GAP	86.7%
+$\mathcal{M}_{Group}$+Bilinear [28]	MLP	87.0%
+$\mathcal{M}_{Group}$+$\mathcal{M}_{Rel}$ (w/o spatial)	Mean Agg.	88.1%
+$\mathcal{M}_{Group}$+$\mathcal{M}_{Rel}$ (w/o multi-level)	Mean Agg.	88.3%
+$\mathcal{M}_{Group}$+$\mathcal{M}_{Rel}$	Mean Agg.	88.6%
+$\mathcal{M}_{Group}$+$\mathcal{M}_{Rel}$	Static	88.5%
+$\mathcal{M}_{Group}$+$\mathcal{M}_{Rel}$	Graph	89.6%

passing shows a preferable high-performance of 89.6%.

What makes a network recognize objects visually? With this question in our mind, we exhibit the visualization results generated by Grad-CAM [33]. The referred three channels of baseline and our model are shown in Fig. 5 (a). It is shown that different channels of our model focus on different object parts, *e.g.*, tail, torso and head of one specific bird. This verifies that constructing inter-channel relationships is beneficial for global object understanding. Despite the noisy regions in baseline (a), the full class activation region of all channels in (b) are always focused on a small specific head region, which greatly restricted the contextual perceiving of objects. In addition, we also present the guided gradient in (c) and (f) for comparison. It shows that the gradient of baseline model focuses on a small region of objects, which would usually lead to overfitting issues.

In Fig. 6, we also present the full activation maps of all channels of three representative models, *i.e.*baseline, trilinear regularizations [49], and our model. It is clear that using higher-order relationship greatly helps the global understanding. Compared to Fig. 6 c), our model generates clearer object boundaries with less noisy regions and provides higher recognition performance.

Table 7. Hyper parameter experiments on graph embedding dim C_N and group num C_r on CUB-200-2011 benchmark.

(C_N,C_r)	(2048,64)	(1024,64)	(512,64)	(512,32)
Acc.	88.7%	89.0%	89.6%	89.1%

Hyperparams. As two main factors in our experiments, the embedding nodes dimension C_N and reduced semantic groups C_r are two main factors that affect the grouping performance. Tab. 7 reveals that constructing higher dimensions nodes with 2048 would be hard for optimization. In experiments, we set $C_N = 512$ and $C_r = 64$ to achieve a performance trade-off with limited data.

5. Conclusions

In this paper, we propose to exploit mix-order relationships in representing fine-grained features with three strategies. The first relation-discovery module exploits the positional enhanced inter-channel relations to form a high-order matrix. Then a graph-based grouping module is proposed to embed this high-order matrix into a low-dimensional manifold. We propose a group-wise training mechanism to update the gradient using group center. Experimental evidence demonstrates the proposed approach achieves new state-of-the-art in fine-grained recognition tasks.

Acknowledgment

This work was supported by grants from National Natural Science Foundation of China (No.61922006), Baidu Academic Collaboration Program and also sponsored by CAAI-Huawei MindSpore Open Fund.

References

[1] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9595–9605, 2019. 7

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR*, 2014. 5

[3] Dongliang Chang, Yifeng Ding, Jiyang Xie, Ayan Kumar Bhunia, Xiaoxu Li, Zhanyu Ma, Ming Wu, Jun Guo, and Yi-Zhe Song. The devil is in the channels: Mutual-channel loss for fine-grained image classification. *IEEE Transactions on Image Processing*, 29:4683–4695, 2020. 2, 3

[4] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5157–5166, 2019. 3, 6, 7

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 6

[6] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6599–6608, 2019. 1, 2, 6, 7

[7] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In *European Conference on Computer Vision (ECCV)*, pages 153–168, 2020. 6

[8] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik. Pairwise confusion for fine-grained visual classification. In *European Conference on Computer Vision (ECCV)*, pages 70–86, 2018. 3, 5, 6, 7

[9] Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Maximum-entropy fine grained classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 637–647, 2018. 3, 6, 7

[10] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4438–4446, 2017. 2, 6

[11] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 317–326, 2016. 3

[12] Yu Gao, Xintong Han, Xun Wang, Weilin Huang, and Matthew Scott. Channel interaction networks for fine-grained image categorization. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 10818–10825, 2020. 2, 4

[13] Weifeng Ge, Xiangru Lin, and Yizhou Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3034–3043, 2019. 1, 2, 6

[14] Abel Gonzalez-Garcia, Davide Modolo, and Vittorio Ferrari. Do semantic parts emerge in convolutional neural networks? *International Journal of Computer Vision*, 126(5):476–494, 2018. 2

[15] Bing He, Jia Li, Yifan Zhao, and Yonghong Tian. Part-regularized near-duplicate vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3997–4005, 2019. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6, 7

[17] Xiangteng He and Yuxin Peng. Weakly supervised learning of part selection model with spatial constraints for fine-grained image classification. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 2

[18] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1173–1182, 2016. 1, 2

[19] Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8662–8672, 2020. 1, 2, 6

[20] Ruyi Ji, Longyin Wen, Libo Zhang, Dawei Du, Yanjun Wu, Chen Zhao, Xianglong Liu, and Feiyue Huang. Attention convolutional binary neural tree for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10468–10477, 2020. 3, 6, 7

[21] Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1062–1071, 2018. 1

[22] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR*, 2017. 4

[23] Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 365–374, 2017. 2, 3

[24] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5546–5555, 2015. 6, 7

[25] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 2, 6

[26] Michael Lam, Behrooz Mahasseni, and Sinisa Todorovic. Fine-grained recognition as hsnet search for informative image parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2520–2529, 2017. 2

[27] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. In *IEEE*

International Conference on Computer Vision (ICCV), pages 2079–2087, 2017. 2

[28] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1457, 2015. 2, 4, 6, 7, 8

[29] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learning for fine-grained visual categorization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8242–8251, 2019. 3, 6, 7

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2, 6

[31] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *European Conference on Computer Vision (ECCV)*, pages 51–66, 2018. 1, 2

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 5

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 4, 8

[34] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1143–1151, 2015. 1, 2

[35] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1857–1865, 2016. 5

[36] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. In *European Conference on Computer Vision (ECCV)*, pages 805–821, 2018. 2, 3, 5, 6, 7

[37] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2015. 2, 6, 7

[38] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 6

[39] Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4148–4157, 2018. 6, 7

[40] Zhihui Wang, Shijie Wang, Shuhui Yang, Haojie Li, Jianjun Li, and Zezhou Li. Weakly supervised fine-grained image classification via guassian mixture model oriented discriminative learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9749–9758, 2020. 3, 6, 7

[41] Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. *Pattern Recognition*, 76:704–714, 2018. 2

[42] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 842–850, 2015. 1, 2

[43] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *European Conference on Computer Vision (ECCV)*, pages 420–435, 2018. 6, 7

[44] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4800–4810, 2018. 5

[45] Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You. Hierarchical bilinear pooling for fine-grained visual recognition. In *European Conference on Computer Vision (ECCV)*, pages 574–589, 2018. 3, 6, 7

[46] Lianbo Zhang, Shaoli Huang, Wei Liu, and Dacheng Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8331–8340, 2019. 2, 3, 7

[47] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision (ECCV)*, pages 834–849. Springer, 2014. 2

[48] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5209–5217, 2017. 1, 6, 7

[49] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5012–5021, 2019. 1, 2, 3, 4, 6, 7, 8

[50] Peiqin Zhuang, Yali Wang, and Yu Qiao. Learning attentive pairwise interaction for fine-grained classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pages 13130–13137, 2020. 2, 6, 7