Weakly Supervised Posture Mining for Fine-grained Classification

Zhenchao Tang[1,†], Hualin Yang[1,†], and Calvin Yu-Chian Chen[1,2,3,*]
[1]Sun Yat-sen University, [2]China Medical University Hospital, [3]Asia University
tangzhch7@mail2.sysu.edu.cn, yanghlin8@mail2.sysu.edu.cn, chenyuchian@mail.sysu.edu.cn

Abstract

Because the subtle differences between the different sub-categories of common visual categories such as bird species, fine-grained classification has been seen as a challenging task for many years. Most previous works focus towards the features in the single discriminative region isolatedly, while neglect the connection between the different discriminative regions in the whole image. However, the relationship between different discriminative regions contains rich posture information and by adding the posture information, model can learn the behavior of the object which attribute to improve the classification performance. In this paper, we propose a novel fine-grained framework named PMRC (posture mining and reverse cross-entropy), which is able to combine with different backbones to good effect. In PMRC, we use the Deep Navigator to generate the discriminative regions from the images, and then use them to construct the graph. We aggregate the graph by message passing and get the classification results. Specifically, in order to force PMRC to learn how to mine the posture information, we design a novel training paradigm, which makes the Deep Navigator and message passing communicate and train together. In addition, we propose the reverse cross-entropy (RCE) and demomenstate that compared to the cross-entropy (CE), RCE can not only promote the accurracy of our model but also generalize to promote the accuracy of other kinds of fine-grained classification models. Experimental results on benchmark datasets confirm that PMRC can achieve state-of-the-art.

1. Introduction

Fine-grained clssification tasks have been seen as quite challenging tasks because the visual differences between the fine-grained classification datasets are hard to recognize. For ordinary people, we can do the normal classification easily, but as for the fine-grained classification, only experts with professional knowledge can do it. Therefore, compared to category-level classification, fine-grained classification is more chanllenging.

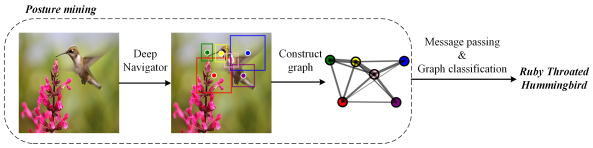

Figure 1. The overview of PMRC. Firstly, we use the Deep Navigator to generate the discriminative regions. Then we construct the graph. Finally, we aggregate the graph through message passing and classify the graph.

There have been many predecessors on fine-grained classification. Works in [2,4,7,12,16,25,31,45,48] can achieve good performance on fine-grained classification. However, their training and testing phase both need bounding box annotations which cost a lot of manual labour and are always error-prone. Then works in [3, 20] develp the methods and use the annotations only in training phase. More recent works develop methods that don't need bounding box annotations in training phase or testing phase [18, 22, 26, 47]. It is a general idea to create graph using local regions. However, related existing method [44, 49] is not easy to transplant, and it is difficult to perceive discriminative regions with correct context information and the relationship between regions. We propose a method that can be conveniently combined with different backbones, and propose a novel learning strategy to ensure that the model can perceive the correct discriminative regions and their relationships (posture information). In addition, our RCE is simple to implement and has better performance than CE.

The framework we propose, which we term PMRC (posture mining and RCE), use the Deep Navigator and graph neural network to mine the posture information from the fine-grained images and use RCE to promote the performance. PMRC is able to combine with different backbones to good effect. We design the loss to make PMRC learn the way to mine the posture information from the images, which include guide the Deep Navigator to search the discriminative regions and guide the message passing module to percept the posture information based on the discrimina-

[†]Equal contribution. *Corresponding author.

tive regions. Besides, the reason we use RCE instead of CE is that although CE can be seen as an appropriate loss function for the normal classification, in training phase, for each sample, it focuses on completing the correct classification as much as possible. In network learning phase, it only concentrates on improving the score of positive labels output by softmax layer, while ignoring the information contained in negative labels. Because of the characteristics of fine-grained classification, negative labels which contain subtle inter-class difference information are very significant. Compared with CE, RCE learns the inter-class difference information by reversing the label score of softmax output layer, so that it has a better effect on fine-grained classification. Specifically, our PMRC has three main steps (see in Figure. 1).

The main contributions of this paper are as follows: (1) We propose a simple framework to mine the posture information in fine-grained classification images, our framework is able to combine easily with different backbones to good effect. (2) We design a novel learning strategy. For the posture mining part, the loss of the Deep Navigator and the loss of message passing communicate with each other to make the model learn how to mine the posture information. For the classification part, we use RCE loss function which can effectively learn the inter-class differences of the samples. (3) PMRC can be trained end-to-end without bounding-box/part annotations. We achieve state-of-the-art on commonly used benchmark.

2. Related Work

2.1. Fine-grained classification

The previous studies on fine-grained data can be divided into three types according to the use of the supervised bounding box labels. The first type is the earliest work, which is fully supervised and needs to use bounding box annotations in the whole phase of training and testing. Berg et al. [2] learned large set of discriminative intermediate-level feature to achieve good performance on bird species identification and face verification. The second method is developed from the fully supervised method. Branson et al. [3] proposed a graph-based clustering algorithm to learn a compact pose normalization space. This method only needs to use all the supervised information during training. Diao et al. [8] explored a unified and strong meta-framework for fine-grained visual classification, but still need extra training data. The third method does not need to use extra training data at any stage of training and testing. Yang et al. [47] proposed navigator-teacher-scrutinizer without extra training data. Zhu et al. [51] proposed a dual cross-attention learning algorithm to coordinate with self-attention learning. PIM [6] can be used as a plug-in, but the improvement in accuracy is not high enough. GCL [44] extracted graph from image, but the pipeline is complicated and not easy to transplant. Our method can automatically mine extra data (posture information) only with image-level annotations, and can be transplanted on different backbones.

2.2. Graph-based Classification

Graph neural network can infer and learn from unstructured data. It is widely used in social networks [10], recommendation systems [42], knowledge graph [36], and other fields. Recent studies show that nodes can be selected from images or videos to build graph neural networks. Yan et al. [46] constructed a spatio-temporal graph to realize pose estimation and behavior classification of videos. For fine-grained classification, it is a general idea to create graph using local regions. [44] and [49] extracted graph from image. But they are not easy to transplant. We propose a simple framework that can be conveniently combined with different backbones, and propose a novel learning strategy to ensure that the model can perceive the correct discriminative regions and their relationships (posture information).

3. Methodology

3.1. Network Architecture

The network architecture is shown in Figure. 2. The input of our method is an image. We obtain the feature maps from the backbone as the shallow feature of the image. The Deep Navigator generates the discriminative regions of the object from the image according to the shallow feature expression, which can be used as the unique attributes of the object. We construct the graph structure of the object according to the position of the disciminative regions in the image. Further, we extract features from the discriminative regions, and correspond these features with the node features of the graph structure to form a complete graph data. Graph data not only contains abstract feature information, but also contains the behavior information of the object which is helpful to improve the accuracy of classification. We apply message passing to graph data, fuse the association between nodes, and further extract the features of nodes. Finally, we calculate the average feature expression of all nodes according to the graph structure, and use the classifier to classify the average features of nodes. Additionally, our method can be easily combined with different backbones and yield good classification performance.

3.2. Deep Navigator and Graph Construction

For fine-grained image classification, the differences between object classes are usually tiny, which makes the labeling of object feature areas need more knowledge of experts. Therefore, we hope to find a weakly supervised method to reduce the annotation cost of datasets and make the model automatically learn some unique attributes of various ob-

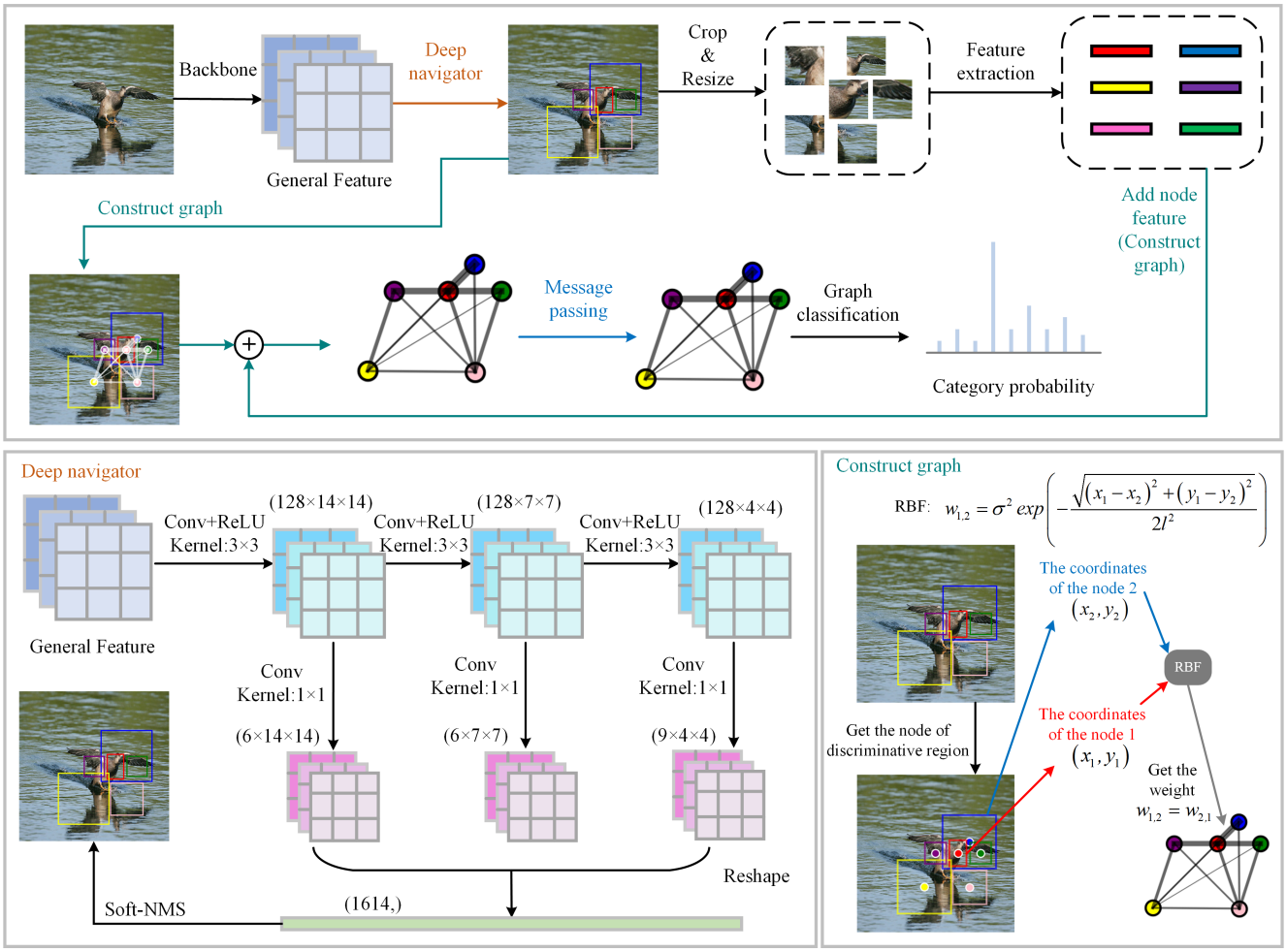

Figure 2. The architecture of PMRC. The PMRC include the Deep Navigator module to generate discriminative regions, the Construct graph module to generate the posture information and the message passing module to fuse the posture information.

jects. Deep Navigator is a lightweight object detection head. We design three different forms of grids in the image to detect discriminative regions with different scales. We use a top-down architecture to detect multi-scale regions through horizontal connection, and use convolutional networks to calculate the hierarchical expression of features layer by layer. Finally, we can get the feature maps at three scales. In the large-scale feature maps, anchors correspond to small regions, and in the small-scale feature maps, anchors correspond to large regions. The architecture of the Deep Navigator is shown in Figure. 2 lower left. The supervised signal comes from the discrminative features through the message passing. The design makes the discriminative regions detected by the Deep Navigator corresponds to the behavior of the object. The implementation details of Deep Navigator can be found in supplementary material section 2.

We use the discrimitavie regions generated by the Deep Navigator to construct the graph of the object. The graph data of the object should be a complete graph. We regard each discriminative region as a node in the graph data, and map the distance between each region into the weight of the edge. We use Gaussian Radial Basis Function(RBF) to express the distance between regions as the weight of edges. The construction process of the graph structure is shown in Figure. 2 lower right. For the features of nodes, we can directly use bilinear interpolation sampling on the feature maps of the backbone to obtain the features of the discriminant region, and compress the features of the discriminant region into vectors through global average pooling. The implementation details of graph construction can be found in supplementary material section 3.

3.3. Message Passing and Graph Classification

The graph of an object contains two types of information. One is the semantic features of the discriminative regions. The other is the spatial association between different

discriminative regions, which is used to express the posture and behavior of objects. We fuse two kinds of information in graph based on message passing, and then classify the graph. We define message passing as:

$$h_{N_{(i)}}^{(l+1)} = aggregate(\{w_{ji} \cdot h_j^{(l)}, \forall j \in N(i)\}) \quad (1)$$

$$h_i^{(l+1)} = sigmoid(W_1 \cdot concat(h_i^{(l)}, h_{N_{(i)}}^{(l+1)})) \quad (2)$$

$$h_i^{(l+1)} = h_i^{(l+1)} / \left\| h_i^{(l+1)} \right\|_2 \quad (3)$$

where $N(\cdot)$ represents the neighbor of the acquisition node, $aggregate(\cdot)$ represents the aggregation mode of the message, $concat(\cdot)$ represents the concate of tensors, and $h^{(l)}$ represents the node features of the layer l. W_1 is a learnable parameter in message passing. Through message passing, we classify the whole graph. Whole graph classification includes graph aggregation and classification of the whole graph. The process can be defined as:

$$score = W_2 \cdot (\frac{1}{|V|} \sum_{v \in V} h_v) \quad (4)$$

where $score$ represents the score of graph data in category, W_2 represents a learnable parameter in the whole graph classification stage, and V is the node set of graph.

3.4. Loss Function and Optimization

First, we record the M discriminative regions by the Deep Navigator as $\{R_1, ..., R_M\}$ and the informativeness corresponding to the M discriminative regions as $\{I_1, ..., I_M\}$. We use the posture and behavior information of the object to guide the Deep Navigator learning. Considering the graph after message passing, we record the features of nodes as $\{h_1, ..., h_M\}$. The global average pooling and sigmoid function are used to convert the features of each node into a value (0-1). We name this value as node confidence, which is used to reflect the probability of the existence of the discriminative regions. We record the node confidence as $\{C_1, ..., C_M\}$, and construct the loss function of the Deep Navigator as:

$$L_{navigator} = \sum_{(m,n)|C_m<C_n} max\{1 - (I_n - I_m), 0\} \quad (5)$$

where $L_{navigator}$ indicates that the higher the node confidence has, the higher informativeness the corresponding discriminative regions have. $L_{navigator}$ helps the model take the posture and behavior information of the object as the supervised signal of the Deep Navigator in training stage.

In order to guide the model to correspond node features, we define CE loss for each discriminative region and aggregate it into message passing loss. The classifier in the whole graph classification stage is used to classify the nodes, and the classification score of the nodes is recorded as s. The node classification score is calculated as:

$$\{s_i\}_{i=1}^M = \{W_2 \cdot h_i\}_{i=1}^M \quad (6)$$

We define the loss of message passing stage as:

$$L_{message} = -\sum_{i=1}^{M} log(\frac{exp(s_i[label])}{\sum_{j=1}^{classnum} exp(s_i[j])}) \quad (7)$$

where $label$ is the real category annotation of the samples and $classnum$ is the number of categories.

$L_{navigator}$ uses the posture and behavior information of the object to guide the Deep Navigator to search the discriminative regions. $L_{message}$ based on discriminative regions guides the massage passing model to perceive the posture and behavior information of the object. The two loss functions promote each other and train together, and make the model learn to mine the posture and behavior information of the object from the image.

Node features lack the global information. In order to add more context information in the process of model reasoning, we obtain the output by backbone, map the features into the scores of categories, and record them as raw. We define the CE loss for the score of features as:

$$L_{backbone} = -log(\frac{exp(raw[label])}{\sum_{j=1}^{classnum} exp(raw[j]))} \quad (8)$$

where $L_{backbone}$ improves the perception ability of the model for global information.

According to the final network classification results $score$, we define RCE for the whole graph classification phase of the training. We invert the network classification results and calculate the probability distribution as:

$$score' = \frac{exp(-score)}{\sum_{j=1}^{classnum} exp(-score[j])} \quad (9)$$

In fact, RCE is simple and reasonable. We reverse the one-hot code of the real category annotations and record the reverse code as R_{label}. In the coding vector, the corresponding values of the real category are 0, the corresponding values of other categories are $1/(classnum - 1)$, and the CE loss after inversion is:

$$L_{graph} = -R_{label}^T log(score') \quad (10)$$

Based on the guidance of L_{graph}, in the reverse results $score'$ of the model, the recognition probability of the correct categories will decrease and the probability of other categories will increase. From the perspective of a single sample, the loss function forces the model to output probability distribution of other categories balanced. From the

perspective of a large number of samples, the probability of correct categories output by the model will also be balanced. This property can help the model reduce intra-class differences and increase inter-class differences, so as to improve the effects of the model on fine-grained classification.

Then, we aggregate these losses and train the model with total loss:

$$L = \alpha L_{backbone} + \beta L_{navigator} + \gamma L_{message} + \theta L_{graph} \tag{11}$$

where $\alpha, \beta, \gamma, \theta$ are used for numerical balance of various losses. Training our method only needs category annotations. We itemize the training steps in supplementary material section4.

4. Datasets and Experiments Configurations

4.1. Datasets

Our experiment datasets include: CUB-200-2011 [39], Stanford Cars [21], FGVC Aircraft [29], Stanford Dogs [19].

CUB-200-2011 It contains 11,788 images of 200 subcategories belonging to birds, 5,994 for training and 5,794 for testing. Each image has detailed annotations: 1 subcategory label, 15 part locations, 312 binary attributes and 1 bounding box. It is generally considered as one of the most competitive datasets since each species has only 30 images for training.

Stanford Cars It consists of 196 classes of cars with a total of 16,185 images taken from the rear. The data is divided into almost a 50-50 train/test split with 8,144 training images and 8,041 testing images. The classes are typically at the level of production year and model.

FGVC Aircraft It contains 10,200 images of aircraft, with 100 images for each of 102 different aircraft model variants, most of which are airplanes. The (main) aircraft in each image is annotated with a tight bounding box and a hierarchical airplane model label.

Stanford Dogs It contains 20,580 images of 120 classes of dogs from around the world, which are divided into 12,000 images for training and 8,580 images for testing.

4.2. Experiments Configurations

We use the Top-1 accuracy to evaluate model's performance. For data preprocessing and hyperparameter settings see supplementary material section5.

In ablation experiment, we set the number of multiple discriminative regions as {3,4,5,6,7,8}. We compare the recognition results and reasoning speed for the graph classification between message passing network and concating the node features directly. We compare the recognition results between using RCE and using CE.

We conduct a large number of model comparison experiments. We applied our method to four datasets: CUB-200-2011, Stanford Cars, Stanford Dogs, FGVC Aircraft, and obtain the recognition results. We compare our method with the best weakly supervised models. In addition, although we do not use any bounding box or part annotations, we still compare with methods which depend on those annotations.

Finally, we conduct more in-depth experiments on RCE. We test the training results of RCE and CE on four datasets. By comparing the training results of a large number of models, the effectiveness of RCE is verified.

5. Results and Discussion

5.1. Ablation study

We analyze the impact of message passing and RCE on the overall architecture. We set the number of multiple discriminative regions as {3,4,5,6,7,8}. We compare the recognition results and reasoning speed for the graph classification between message passing network and concating the node features directly. And we also compare the results between using RCE and using CE.

Table 1. Ablation study with Top-1 accuracy of Message Passing and RCE on CUB-200-2011(%), backbone: ResNet50.

Regions number	3	4	5	6	7	8
Message(RCE)	**90.9**	**91.0**	**91.3**	**91.8**	**91.5**	**91.4**
Concat(RCE)	88.6	88.9	89.3	89.1	89.0	89.1
Message(CE)	**89.2**	**89.4**	**90.1**	**90.6**	**90.4**	**90.4**
Concat(CE)	87.3	87.4	87.6	87.6	87.3	87.5

Table 2. Reasoning Speed and Accuracy Test on CUB-200-2011 (Top-1 Accuracy)

Method	backbone	Speed (fps)	Accuracy(%)
ResNet50 (RCE)	-	91	85.7
ResNet50 (CE)	-	91	84.5
Message+6 Regions (RCE)	ResNet50	85	91.8
Message+7 Regions (RCE)	ResNet50	84	91.5
Message+6 Regions (CE)	ResNet50	85	90.6
Message+7 Regions (CE)	ResNet50	84	90.4
SwinTrans (RCE)	-	49	91.2
SwinTrans (CE)	-	49	90.3
Message+6 Regions (RCE)	SwinTrans	45	94.3
Message+6 Regions (CE)	SwinTrans	45	**93.5**

As can be seen from Table 1, the overall results of using the message passing mechanism is better than that of directly concating the discriminative region features. Message passing integrates the behavior information of the object, which improves the recognition effect of the model on fine-grained images. In addition, the model trained with RCE is also better than the model trained with CE. RCE forces the model to further learn the inter-class differences

of fine-grained objects, and improves the recognition ability of the model for difficult classification objects.

We test the reasoning speed of PMRC. The results are shown in Table 2. Based on Table 1, we select the first two settings with the highest accuracy: 6 regions and 7 regions, and compare the reasoning speed and accuracy. It can be seen from Table 2 that PMRC only adds a small amount of extra time overhead to the backbone, the accuracy is much higher than its backbone. Importantly, PMRC (SwinTransformer) trained only with CE has achieved state-of-the-art on CUB-200-2011 (compare with Table 3).

The results of ablation experiments on other datasets and the results of different loss ratios $(\alpha, \beta, \gamma, \theta)$ can be found in supplementary material section6.

5.2. Comparisons with existing approaches

Our comparisons focus on the weakly supervised methods because the proposed model only utilizes image-level annotations. Table 3 shows the performance of different methods on CUB-200-2011, Stanford Cars, FGVC Aircraft, Stanford Dogs. In Table IV from top to bottom, the methods are separated into four groups, which are (1) supervised methods, (2) recent weakly supervised methods, (3) backbones, (4) our method PMRC.

Strong supervised methods rely on the object and even part annotations to achieve comparable results. However, using the object or part annotations limits the performance due to the fact that human annotations only give the coordinates of important parts rather than the accurate discriminative region location. Weakly supervised methods gradually exceed the strong supervised methods though picking out discriminative regions. PMRC outperforms the strong supervised methods such as DATL and MetaFormer, showing the importance of PMRC for discriminative feature learning. PMRC outperforms the weakly supervised methods such as PIM and DCAL. This shows that we can make the recognition result surpass the latest method based on vision transformer by introducing posture information to fine-grained tasks. Compared with API net, PMRC does not need to build image pairs based on datasets. PMRC achieves better classification effect by learning the posture information in a single image. GCL tries to learn the association of different regions in the image and achieves good recognition effect. PMRC further mines the posture information hidden in the image, and uses RCE to improve the ability of the model to learn the inter-class differences. Therefore, the recognition effect of PMRC is obviously better than GCL. NTS-Net only considers the prediction of discriminative regions and ignores the correlation between different regions. Therefore, the recognition effect of PMRC is better than NTS-Net. PMRC has achieved excellent recognition results on four widely used datasets.

Separated from the complex backbone, the method of learning the posture information from images and using RCE to increase the differences of heterogeneous objects can be widely applied to the tasks related to fine-grained classification. PMRC can be easily combined with different backbones and get good fine-grained recognition performance.

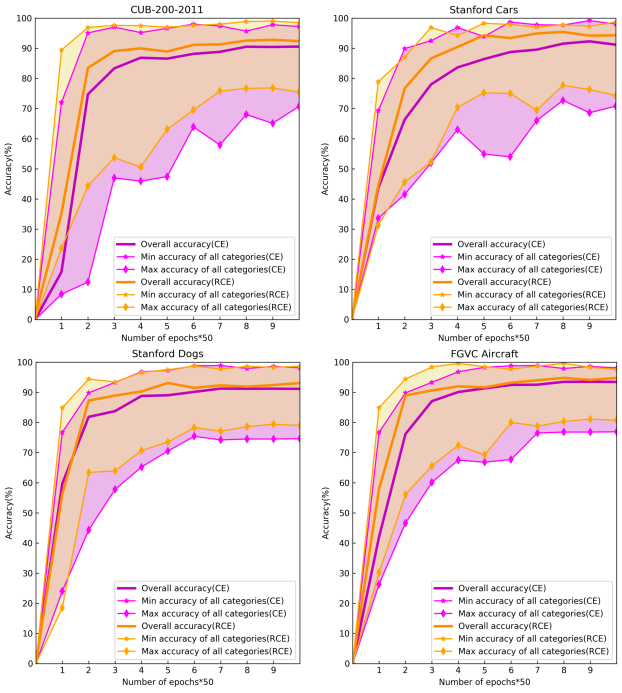

Figure 3. Comparison of training effects between CE and RCE. We compare the effectiveness of CE and RCE on the four datasets (CUB-200-2011, Stanford Cars, Stanford Dogs, FGVC Aircraft). After every 50 epochs, we test Top-1 accuracy on the test datasets.

5.3. RCE for Fine-grained Classification

We have verified the excellent performance of RCE in fine-grained classification through a large number of experiments. In Figure. 3, we visualize the training effects of the PMRC on the four datasets (CUB-200-2011, Stanford Cars, Stanford Dogs, FGVC Aircraft) to compare the effectiveness of CE and RCE. We set the model to train 500 epochs. After every 50 epochs, we use the model for testing, and record the classification accuracy of each category. It can be clearly seen in Figure. 3 that the minimum accuracy line of CE is always under the counterpart of RCE in all the four datasets, which means that for the category which is the most difficult to classify, the effectiveness of using RCE is better than use CE. And the maximum accuracy line of RCE is always above the counterpart of CE, which means that for the category which is not difficult to classify, the effective of using RCE is better than CE. As for the overall accuracy, RCE is also better than CE. So, we can draw a

Table 3. Comparison of Different Methods on CUB-200-2011, Stanford Cars, FGVC Aircraft and Stanford Dogs. (Top-1 Accuracy(%))

Method	Extra Supervision	CUB-200-2011	Stanford Cars	FGVC Aircraft	Stanford Dogs	speed	params
MetaFormer [8]	✓	**92.9**	**95.4**	92.8	-	-	-
DATL [17]	✓	91.2	94.5	**93.1**	**92.2**	-	-
TA-FGVC [23]	✓	88.1	-	-	88.9	-	-
PA-CNN [20]	✓	85.4	92.8	-	-	-	-
BoT [43]	✓	-	92.5	88.4	-	-	-
FCAN [27]	✓	84.3	91.3	-	88.9	-	-
MG-CNN [40]	✓	85.1	-	86.6	-	-	-
PIM [6]	×	**92.8**	-	-	-	-	-
DCAL [51]	×	92.0	95.3	93.3	-	-	-
Vit-SAC [9]	×	91.8	94.5	93.1	-	-	-
CAP [1]	×	91.8	-	**94.9**	-	-	-
TransFG [13]	×	91.7	94.8	-	**92.3**	-	-
FFVT [41]	×	91.6	-	-	91.5	-	-
CAL [33]	×	90.6	**95.5**	94.2	-	-	-
Inception-v4 [32]	×	-	95.3	-	-	-	-
API-Net [52]	×	90.0	95.3	-	90.3	-	-
DenseNet161+MM+FRL [50]	×	88.5	95.2	-	-	-	-
GCL [44]	×	88.3	95.1	93.2	90.5	-	-
NTS-Net [47]	×	87.5	91.4	93.9	-	-	-
SwinTransformer [28]	×	90.3	92.7	90.6	91.1	49fps	88M
ResNet50 [24]	×	84.5	88.6	87.2	84.7	91fps	25M
DenseNet161 [15]	×	86.6	90.4	90.9	88.3	38fps	29M
VGG16 [37]	×	77.8	83.3	85.3	81.6	68fps	138M
Our PMRC (SwinTransformer)	×	**94.3**	**96.9**	**96.7**	**95.2**	45fps	89M
Our PMRC (ResNet50)	×	91.8	95.4	94.8	93.1	85fps	26M
Our PMRC (DenseNet101)	×	91.3	95.2	94.0	92.7	36fps	30M
Our PMRC (VGG16)	×	86.3	89.1	91.3	89.9	63fps	139M

Table 4. Comparison of CE and RCE in previous fine-grained tasks with the increment of Top-1 accuracy(%).

Method	Extra.S	CUB	Cars	Aircraft	Dogs
ResNet50 [14]	×	+1.23	+0.62	+0.54	+1.16
DenseNet161 [15]	×	+0.37	+0.70	+1.09	+0.71
Xception [5]	×	+1.17	+1.52	+1.19	+1.60
Incep.V3 [38]	×	+1.11	+1.27	+1.73	+1.82
MobileNetV2 [35]	×	+1.13	+1.69	+1.04	+1.28
B-CNN [26]	×	+1.68	+0.97	+0.99	+1.42
NTS-Net [47]	×	+1.72	+1.60	+2.12	+1.45
DATL [17]	✓	-0.28	+0.27	-0.19	+0.11
DAT [30]	✓	+1.20	+1.25	+1.53	+1.25
TResNet-L-V2 [34]	✓	-0.07	+0.42	+0.18	+0.36
SAM [11]	✓	+1.12	+1.68	+1.96	+1.65
MG-CNN [40]	✓	+1.62	+1.28	+1.02	+1.41

conclusion that, for fine-grained classification, RCE is more suitable compared to CE. Besides, we visualize the embedding (learned from CE and RCE, respectively) with t-sne in supplementary material section7.

We use RCE to the fine-grained classification models and test the recognition effect. Record the change of Top-1 accuracy compared with CE training, see Table 4. From Table 4, it can be seen that RCE improves the test performance of various algorithms on four datasets as a whole. There are only a few cases where the recognition results decline, but the degree of decline is very small compared with the degree of improvement. Therefore, we can believe that RCE can maintain or improve the recognition effect of these algorithms. On the four benchmark datasets, a large number of algorithms have improved their performance, which proves that RCE has good generalization. Therefore, for fine-grained image classification tasks, we can use RCE as a simple and practical loss function.

5.4. Discussion of Posture Mining

For fine-grained image classification, it is meaningful for us to mine posture information in images. It can be explained from the following view:

It is not comprehensive to simply consider the features of discriminative regions. For example, for the fine-grained classification of animals, consider a common phenomenon: the hair on the head of one kind of animal is very similar to the abdominal hair of another kind of animal. Suppose that we get a region in which the content is hair, if there is no posture information from space, the model cannot distinguish whether it is head hair or abdomen hair, which is easy to cause errors in the classification results of objects. If we introduce posture information, the model can accurately distinguish head hair from abdominal hair, which will further improve the ability of the model to perceive the differences between different categories. Therefore, the mining

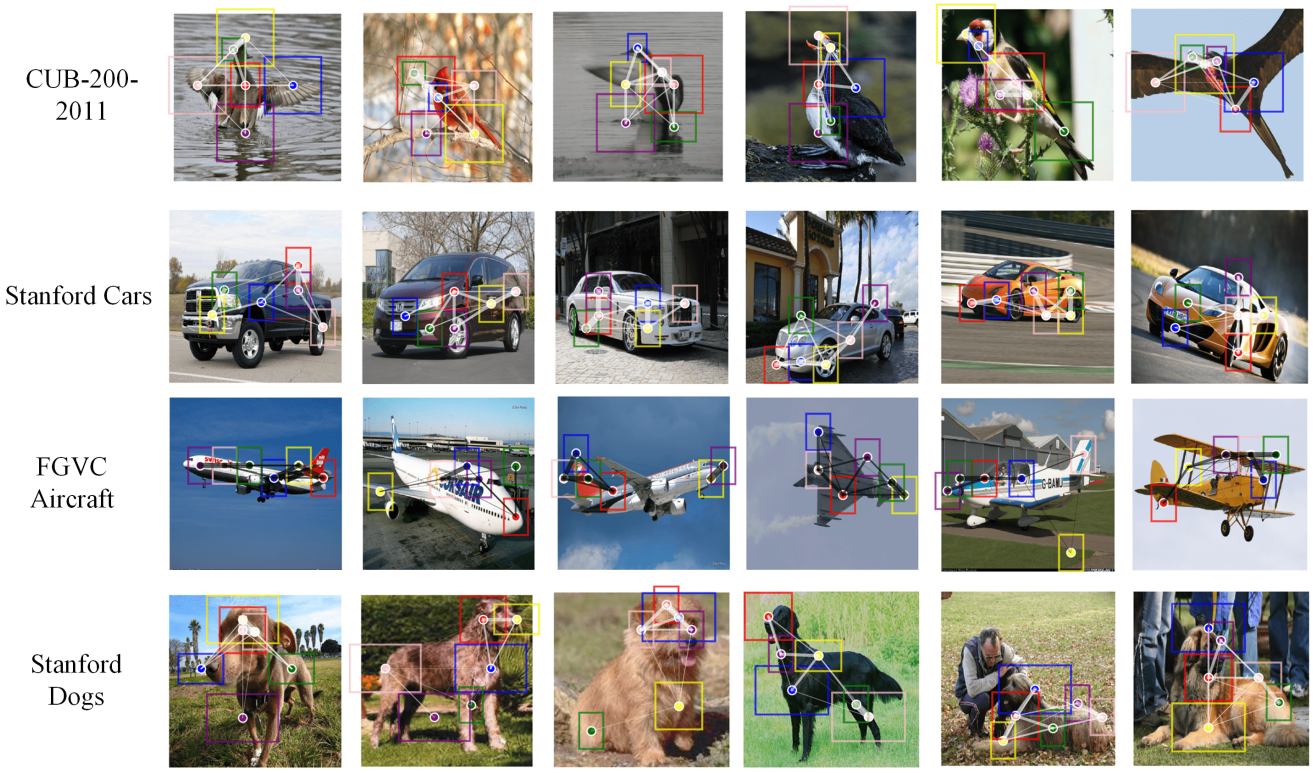

Figure 4. Based on four datasets, the visualization of the discriminative regions and posture. The first row to the fourth row correspond to CUB-200-2011, Stanford Cars, FGVC Aircraft, Stanford Dogs. Bounding boxes represent the discriminative regions of the object, and graph represents the posture of the object.

of posture information is meaningful to image classification.

For this view, we visualize the discriminative regions and posture information of four datasets, as shown in Figure. 4. PMRC can extract discriminative regions from the image and roughly estimate the posture information of the object. Based on posture information, we can get a complete expression with spatial context information by using message passing and fusing the features of discriminative regions. For birds and dogs, their posture information describes the relationship between head, abdomen and tail. For cars and aircrafts, their posture information comes from their mechanical structure.

Posture information is essentially the spatial relationship between discriminative regions. The posture information is useful, because in the process of PMRC learning the posture information, it can promote the model to capture the correct context information, so as to obtain the correct discriminant region closely related to the object. This is an adaptive detection of discriminant regions. Existing methods do not attempt to learn posture information, which leads to discriminative regions found by existing methods that may contain useless contextual information. Therefore, PMRC always has excellent performance by combining with different backbones.

6. Conclusion

In this paper, we propose a novel method for fine-grained classification by introducing the posture information. PMRC is a framework that can be combined with different backbones and is trained by weakly supervised signals. We claim three major contributions. First, we propose a simple module to mine the posture information, the module is able to combine conveniently with existing backbones. Second, we design a novel learning strategy to force PMRC to learn how to mine the posture information. Third, we demonstrate that compared to using traditional CE loss, RCE loss is more suitable to fine-grained classification tasks. Combining the above approaches produces PMRC which achieve state-of-the-art on four benchmark datasets.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 62176272) and China Medical University Hospital (DMR-112-085).

References

[1] Ardhendu Behera, Zachary Wharton, Pradeep Hewage, and Asish Bera. Context-aware attentional pooling (cap) for fine-grained visual classification. *arXiv preprint arXiv:2101.06635*, 2021. 7

[2] Thomas Berg and Peter N Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 955–962, 2013. 1, 2

[3] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014. 1, 2

[4] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 321–328, 2013. 1

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 7

[6] Po-Yung Chou, Cheng-Hung Lin, and Wen-Chung Kao. A novel plug-in module for fine-grained visual classification. *arXiv preprint arXiv:2202.03822*, 2022. 2, 7

[7] David Cossock and Tong Zhang. Statistical analysis of bayes optimal subset ranking. *IEEE Transactions on Information Theory*, 54(11):5140–5154, 2008. 1

[8] Qishuai Diao, Yi Jiang, Bin Wen, Jia Sun, and Zehuan Yuan. Metaformer: A unified meta framework for fine-grained recognition. *arXiv preprint arXiv:2203.02751*, 2022. 2, 7

[9] Tuong Do, Huy Tran, Erman Tjiputra, Quang D Tran, and Anh Nguyen. Fine-grained visual classification using self assessment classifier. *arXiv preprint arXiv:2205.10529*, 2022. 7

[10] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426, 2019. 2

[11] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. 7

[12] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. In *Proceedings of the IEEE international conference on computer vision*, pages 1713–1720, 2013. 1

[13] Ju He, Jie-Neng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, Changhu Wang, and Alan Yuille. Transfg: A transformer architecture for fine-grained recognition. *arXiv preprint arXiv:2103.07976*, 2021. 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 7

[16] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1173–1182, 2016. 1

[17] Ashiq Imran and Vassilis Athitsos. Domain adaptive transfer learning on visual attention aware data augmentation for fine-grained visual categorization. In *International Symposium on Visual Computing*, pages 53–65. Springer, 2020. 7

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015. 1

[19] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2. Citeseer, 2011. 5

[20] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5546–5555, 2015. 1, 7

[21] 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. 5

[22] Michael Lam, Behrooz Mahasseni, and Sinisa Todorovic. Fine-grained recognition as hsnet search for informative image parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2520–2529, 2017. 1

[23] Jingjing Li, Lei Zhu, Zi Huang, Ke Lu, and Jidong Zhao. I read, i saw, i tell: Texts assisted fine-grained visual classification. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 663–671, 2018. 7

[24] Zhichao Li, Yi Yang, Xiao Liu, Feng Zhou, Shilei Wen, and Wei Xu. Dynamic computational time for visual attention. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1199–1209, 2017. 7

[25] Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1666–1674, 2015. 1

[26] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015. 1, 7

[27] Xiao Liu, Tian Xia, Jiang Wang, Yi Yang, Feng Zhou, and Yuanqing Lin. Fully convolutional attention networks for fine-grained recognition. *arXiv preprint arXiv:1603.06765*, 2016. 7

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021. 7

[29] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[30] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018. 7

[31] Omkar M Parkhi, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. The truth about cats and dogs. In *2011 International Conference on Computer Vision*, pages 1427–1434. IEEE, 2011. 1

[32] Jo Plested, Xuyang Shen, and Tom Gedeon. Non-binary deep transfer learning for image classification. *arXiv preprint arXiv:2107.08585*, 2021. 7

[33] Yongming Rao, Guangyi Chen, Jiwen Lu, and Jie Zhou. Counterfactual attention learning for fine-grained visual categorization and re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1025–1034, 2021. 7

[34] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 7

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 7

[36] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018. 2

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 7

[39] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[40] Dequan Wang, Zhiqiang Shen, Jie Shao, Wei Zhang, Xiangyang Xue, and Zheng Zhang. Multiple granularity descriptors for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision*, pages 2399–2406, 2015. 7

[41] Jun Wang, Xiaohan Yu, and Yongsheng Gao. Feature fusion vision transformer for fine-grained visual categorization. *arXiv preprint arXiv:2107.02341*, 2021. 7

[42] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019. 2

[43] Yaming Wang, Jonghyun Choi, Vlad Morariu, and Larry S Davis. Mining discriminative triplets of patches for fine-grained classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1163–1172, 2016. 7

[44] Zhuhui Wang, Shijie Wang, Haojie Li, Zhi Dou, and Jianjun Li. Graph-propagation based correlation learning for weakly supervised fine-grained image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12289–12296, 2020. 1, 2, 7

[45] Lingxi Xie, Qi Tian, Richang Hong, Shuicheng Yan, and Bo Zhang. Hierarchical part matching for fine-grained visual categorization. In *Proceedings of the IEEE international conference on computer vision*, pages 1641–1648, 2013. 1

[46] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 2

[47] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 420–435, 2018. 1, 2, 7

[48] Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1143–1152, 2016. 1

[49] Yifan Zhao, Ke Yan, Feiyue Huang, and Jia Li. Graph-based high-order relation discovery for fine-grained recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15079–15088, 2021. 1, 2

[50] Runkai Zheng, Zhijia Yu, Yinqi Zhang, Chris Ding, Hei Victor Cheng, and Li Liu. Learning class unique features in fine-grained visual classification. *arXiv preprint arXiv:2011.10951*, 2020. 7

[51] Haowei Zhu, Wenjing Ke, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Dual cross-attention learning for fine-grained visual categorization and object re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4692–4702, 2022. 2, 7

[52] Peiqin Zhuang, Yali Wang, and Yu Qiao. Learning attentive pairwise interaction for fine-grained classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13130–13137, 2020. 7