Neurocomputing 453 (2021) 26–37

Contents lists available at ScienceDirect

Neurocomputing

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

Brief papers

Hard Decorrelated Centralized Loss for fine-grained image retrieval

Xianxian Zeng [a,1], Shun Liu [b,1], Xiaodong Wang [c,*], Yun Zhang [c,*], Kairui Chen [d], Dong Li [c]

[a] School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510000, China
[b] School of Automation, Guangdong Polytechnic Normal University, Guangzhou 510000, China
[c] Automation, Guangdong University of Technology, Guangzhou 510006, China
[d] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 6 April 2020
Revised 12 February 2021
Accepted 8 April 2021
Available online 13 April 2021
Communicated by Zidong Wang

2010 MSC:
00-01
99-00

Keywords:
Fine-grained image retrieval
Hard Decorrelated Centralized Loss
Convolutional neural network

ABSTRACT

Although there is abundant of investigations on fine-grained image retrieval, it is still an extremely challenging task in the field of computer vision, due to the character of small diversity in inter-class but large diversity within intra-class. To handle this task, loss functions are critical to the performance of a deep convolutional neural network in extracting the discriminative feature of the fine-grained image for retrieval. Recent studies showed that the global structure loss functions help to extract more discriminative features. In this paper, we introduce a novel global structure loss function, named Hard Decorrelated Centralized Loss, for further improving the representation for fine-grained image retrieval. The proposed loss is available in extracting the discriminative feature for dividing the most similar categories. In our experiments, we employ the proposed loss to train the convolutional neural network, which shows state-of-the-art performances on six classical fine-grained image retrieval benchmarks, e.g. CUB-200-2011 and Stanford Cars.

1. Introduction

In Fine-grained image retrieval (FGIR), given database images of the same species, e.g. birds [1], cars [2] or dogs [3], and a query order, the machine should return images of the same subspecies of the query. FGIR is a challenging task in computer vision and attracts increasing research focus [4–7]. Different from general image retrieval tasks that focus on retrieving the near-duplicate images based on the contents like colors and shapes, the target of FGIR is to retrieve the same type of images. As shown in Fig. 1, images of intra-class possess large differences, e.g. pose, illumination, viewpoint and background (emphasized by the red dotted rectangle), but images among different classes are similar (emphasized by the green dotted rectangle). Therefore, FGIR needs to extract discriminative features to distinguish fine-grained categories, which makes it more difficult than the ordinary image retrieval task.

Recently, the successful applicationof convolutional neural networks have driven significant advances in computer vision and image understanding, such as AlexNet [8] VGGNet [9] ResNet [10] for image classification, web page categorization [11] and object detection [12]. Series of methods [13,14] used CNN with metric learning losses to extract distinguishing features and achieved better performances of FGIR. Earlier works in FGIR mainly based on local structure losses, encoding the example relationship locally. Pairwise loss [13] and triplet loss [15] were two prevalent-local structure losses, but both of them fall short in accuracy results. To overcome this problem, a weakly-supervised localization method [5] and the global structure loss functions [6,7] were proposed. Wei et al. proposed a unsupervised method named SCDA [5] for FGIR, which employed the last feature map to select the discriminative region, and then utilizes the combination of max pooling and average pooling to extract discriminative features. Zheng et al. first developed the global structure loss named Centralized Ranking Loss (CRL) [6], which optimized the intra-class compactness and inter-class separability in a global way. And CRL-WSL [6] was an improving version of CRL with weakly supervised localization. Later, Decorrelated Global Centralized Ranking Loss (DGCRL) [7] lessens the gap between euclidean distance and cosine similarity by the Norm-Scale layer. These methods enhanced models retrieval performance and promoted the development of FGIR.

In Fig. 1, on the one hand, four columns of Flycatchers (Sparrows) are much more similar for they belong to one of the subspecies of birds, thus would be hard to identify. On the other

* Corresponding authors.
E-mail addresses: 2111604062@mail2.gdut.edu.cn (X. Wang), yun@gdut.edu.cn (Y. Zhang).
[1] Xianxian Zeng and Shun Liu have contributed equally to this work.

https://doi.org/10.1016/j.neucom.2021.04.030
0925-2312/

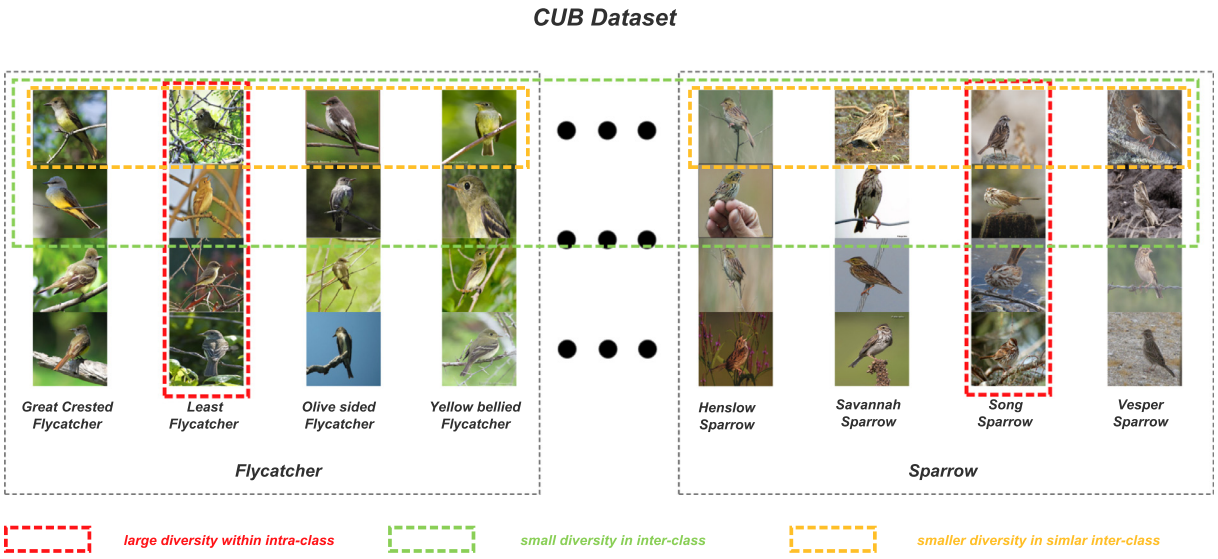


Fig. 1. Illustration of Flycatcher and Sparrow in dataset CUB-200-2011 . There are four classes of bird subspecies Flycatchers and Sparrows respectively. Given these eight columns (classes) of birds, it is natural to identify Flycatcher and Sparrow (inter-class), e.g. Great Crested Flycatcher and Henslow Sparrow, while machine usually gets stuck in retrieving images of categories in the similar classes, e.g. Great Crested Flycatcher and Least Flycatcher (emphasized by the left orange dotted rectangle), because of the analogous appearance and texture.

hand, the first class (Great Crested Flycatcher) and the seventh class (Song Sparrow) in Fig. 1 are obviously different in the perspective of shape and feather pattern, which could be distinguished readily. Bear in mind that the performance of FGIR would be improved if the extracted features can effectively separate much closer subspecies (emphasized by the left orange dotted rectangle in Fig. 1, e.g. Great Crested Flycatcher and Yellow-bellied Flycatcher or Henslow Sparrow and Savannah Sparrow). Intuitively, we believe that if incorporating hard mining ideas into the global loss function in training period to distinguish closet categories, the retrieval capacity of models would be improved. Therefore, in this paper, based on the global structure loss [7], we develop a novel softmax loss function with hard mining strategy to extract more distinguishing features in FGIR. The proposed loss function aims at differentiating the top k closest subspecies via mining their global central featuresto improve the FGIR performance. Fig. 2 vividly depicts the difference between general global loss and our proposed loss. Extensive experiments verify that models trained with our loss function achieve better performance thanstate-of-the-art methods in terms of six popular FGIR benchmarks such as Stanford Cars [2] and CUB-200-2011 [1].

Main contributions of this paper can be summarized as follows:

1. A novel variant of softmax loss based on hard miningis proposed to train the network, whose capacity ofseparating the closest (similar) subspecies could be further improved.
2. The proposed loss function can be directly applied in training various CNN models to improve FGIR performances.
3. Extensive experiments on several classical benchmarks demonstrate the effectiveness of our proposed loss function in improving FGIR and FGOR (fine-grained object recognition) performances, respectively.

The rest of this paper is organized as follows. Section 2 introduces the related works of FGIR. Section 3 describes the problem definition, overview of FGIR, and the proposed loss. Details of experiments including hyper-parameters adjustments and results analysis are shown in Section 4. We arrive at discussing the influence of the proposed loss in Section 5.

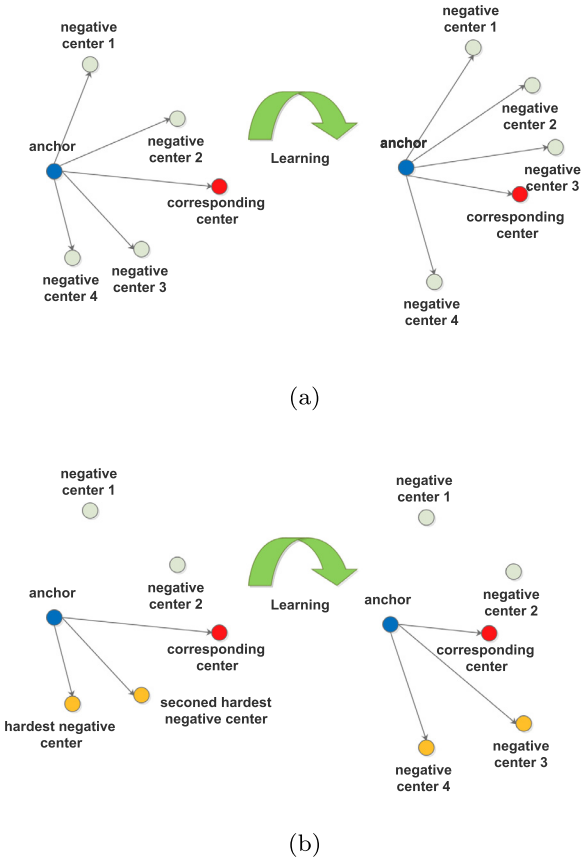


Fig. 2. The diagrams of general global losses and the proposed loss. Here we define the input feature as the anchor. (a) The general global loss lessens the distance between an anchor and its corresponding center, and enlarge the distance among the anchor and other class centers simultaneously. (b) The proposed loss lessens the distance between an anchor and its corresponding center, and enlarges the distances between among anchor and its top k closest negative centers..

2. Related work

In this section, we present a brief history of general image retrieval, deep metric learning and FGIR.

Image retrieval is one of the subjects of concern for computer vision. Given an input query, image retrieval is to search similar images of the same class in a large, unordered, or even chaotic collection of images. Therefore, the image retrieval system needs to be capable of distinguishing features thatrepresent the input images. In the early stage of image retrieval, a series of image retrieval methods were mainly based on local features (e.g. SIFT [16]) and feature aggregation strategies. Nearly a decade later, Fisher Vector (FV) [17] and Vector of Locally Aggregated Descriptors (VLAD) [18] were proposed and both of them were considered as representative feature aggregation methods. They used the local descriptor to extract a group of features at the beginning, then the fisher kernel, or a simplified version of which is utilized to transform the incoming group of features into a fixed size discriminative vector representation.

As CNN becoming a success in image classification, some deep-learning-based methods were proposed for image retrieval, and they obtained prominent and satisfactory results. Gong et al. [19] developed the multi-scale orderless pooling to improve the invariance of CNN activations while keeping the discriminative ability. Babenko et al. [20] proposed the sum-pooled convolutional features to obtain better performance. Tolias et al. [21] defined a compact image representation derived from the convolutional layer activations. The representation encodes multiple image regions without the need to refeed multiple inputs to the network. Noh et al. [22] derived an attentive local feature descriptor for large-scale image retrieval. Radenovic et al. [23] used reconstructed 3D models to select the training data, which further enhance performance of fine-tune CNNs for image retrieval.

Deep metric learning aims at minimizing the distance among similar images, and is an alternative way to improve the retrieval. Pairwise loss [13] and triplet loss [15] were the most famous loss functions for deep metric learning. However, they suffered from slow convergence. To solve this problem, Sohn [24] proposed a multi-class N-pair loss to improve upon the triplet loss. Ustinova et al. [25] defined a loss without tuning parameters, named Histogram loss, for learning deep embeddings. Song et al. [14] developed a high-order similarity constraint based on the lifted pairwise distance matrix within the mini-batch. Huang et al. [26] derived the Position-Dependent Deep Metric (PDDM) unit for learning a similarity metric.

Fine-grained image retrieval aims to differentiate subordinate classes in the task of image retrieval. Zhang et al. [27] used a multi-task learning framework to learn fine-grained feature representations. Wei et al. [5] employed pre-trained CNN models to obtain the deep descriptors by selecting the important region of the main object unsupervisedly. Zheng et al. [6] proposed CRL to speedup training CNN for FGIR. Furthermore, Zheng et al. found that the previous works in FGIR were trained with local structure loss functions like triplet loss and its variants. Therefore, Zheng et al. proposed the Normalized-Scale Layer and a global structure loss DGCRL [7], to promote the performance in FGIR. To further enhance the feature extraction capacity of CNN models in FGIR, we propose a new global structure loss function, named Hard Decorrelated Centralized Loss, to train the models.

3. The proposed method

As shown in Fig. 3, the pipeline of our method contains two parts: feature extraction and the training period. Similar to previous works [6,7], for the sake of fairness in subsequent comparison experiments, we select ResNet structure as the CNN backbone for feature extraction. In the training period, we apply the novel global structure loss, named Hard Decorrelated Centralized Loss, to train the model. The obvious difference between our loss and previous methods is that our loss selects top k hardest negative global centers for training and hence allows the extracted features become more discriminate.

3.1. Problem definition and overview

FGIR is a task to return images of the same class of the query in the testing fine-grained image database. Generally, these fine-grained image databases contain a training set and a testing set. A CNN model is proposed to project input images to embedding vectors, which is employed to compute the distance for image retrieval. Commonly, models are trained on the training set and then evaluated on the testing set. In the training stage, each image $\mathbf{I}_i$ in the training set $\mathscr{I} = \{(\mathbf{I}_1, y_1), (\mathbf{I}_2, y_2), \ldots, (\mathbf{I}_N, y_N)\}$ has a corresponding label $y_i, i = 1, \cdots, N$. Here, through the training data, i.e. $\mathscr{I}$, the CNN model can be defined as:

$$\mathbf{f}_i = F(\mathbf{I}_i, \theta), \tag{1}$$

where parameter θ will be automatically tuned after training and $F(\mathbf{I}_i, \theta)$ captures discriminative feature $\mathbf{f}_i \in R^d$ (each image can be transformed to d dimensional features) for image retrieval. Therefore, it is of crucial importance to design a loss function to train the CNN model. From [7], the existing loss functions for training in FGIR can be categorized into two groups: the local structure loss and the global structure loss.

Local structure loss defines that the feature relationship is encoded inside the training batch. Pairwise, triplet and quadruplet loss are local structure loss functions. Specifically, given a triplet data $\{\mathbf{I}_a, \mathbf{I}_p, \mathbf{I}_n\}$, where $\mathbf{I}_a$ and $\mathbf{I}_p$ are two images from the same class while $\mathbf{I}_n$ is an image of another class, then the model extracts a triplet feature $\{\mathbf{f}_a, \mathbf{f}_p, \mathbf{f}_n\}$. After that, the triplet loss function [15] can be computed as

$$L_{triplet} = \frac{1}{2}\max(0, m + \mathbf{D}(a, p) - \mathbf{D}(a, n)), \tag{2}$$

where $\mathbf{D}(a, p) = \|\mathbf{f}_a - \mathbf{f}_p\|_2$ is the positive distance between $\mathbf{f}_a$ and $\mathbf{f}_p$, $\mathbf{D}(a, n) = \|\mathbf{f}_a - \mathbf{f}_n\|_2$ is the negative distance between $\mathbf{f}_a$ and $\mathbf{f}_n$, m is the margin value. Apparently, the target of 2 is to minimizing $\mathbf{D}(a, p)$ and maximizing $\mathbf{D}(a, n)$. Experiments in [28] show that the sampling strategy for triplet data plays an important role in achieving better performance. However, this local scope leads to vast search space and low accuracy. Besides, the triplet loss function needs large mini-batches for training. To overcome this issue, the global structure loss function is proposed.

Global structure loss defines the feature relationship in a global way. Assume that there were K classes/clusters in the training set, corresponding to K class centers, denoted by $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$. Thus, Zheng et al. [6] constructed CRL in terms of K centers, i.e. $\mathbf{C}$, to train the CNN model.

$$L_{\mathrm{CRL}} = \sum_{\mathbf{c}_l \in C, l \neq k} \sum_{i} \max(0, m + \|\mathbf{f}_i - \mathbf{c}_k\|_2 - \|\mathbf{f}_i - c_l\|_2), \tag{3}$$

where $\mathbf{c}_k$ is the class center of $\mathbf{f}_i$ while $\mathbf{c}_l$ is the class center differ from $\mathbf{f}_i$. CRL conducts the extracting features $\mathbf{f}_i$ to get close the center of target class and deviate from other class centers. Therefore, the performance of CRL does not depend on the sampling strategy. Comparing to triplet loss, CRL achieves nearly 1000 times speedup but better performance.

Based on softmax loss, one of the global structure loss function, significant improvements have been achieved in image classification [10,29]. Here, weights of the last fully connected layer

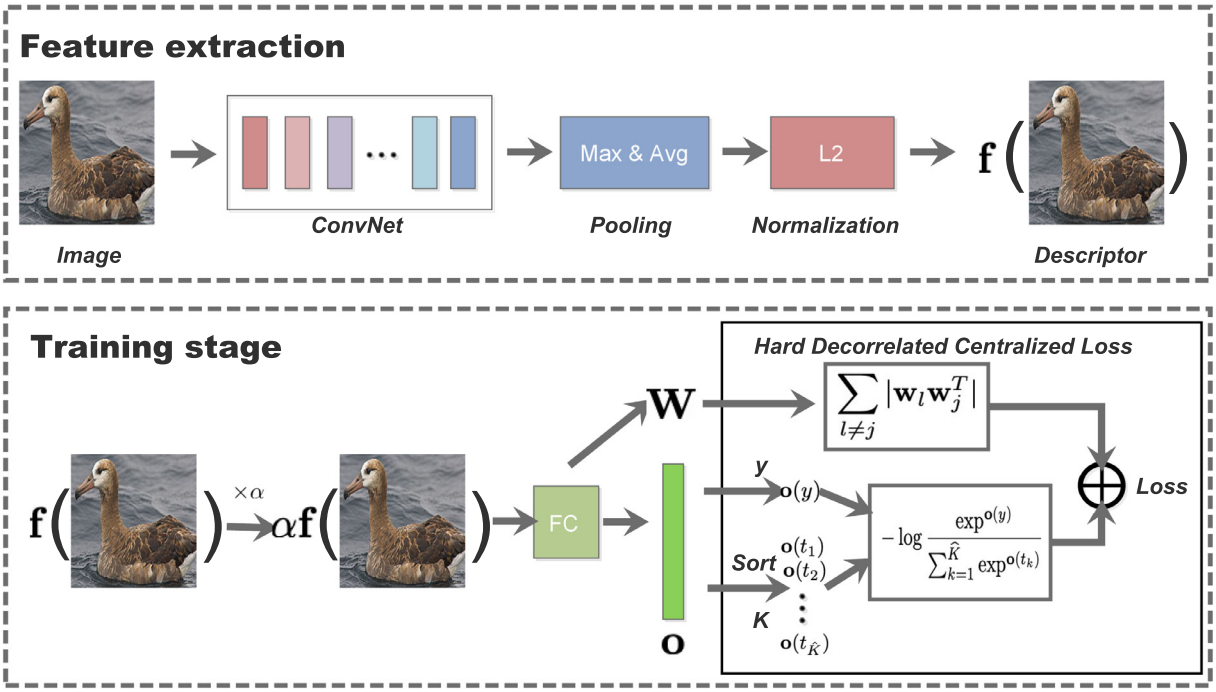


Fig. 3. The pipeline of our method. The feature extraction module is a CNN backbone with the combinative output of max pooling and average pooling. And we employ the proposed loss function to train the feature extraction module. In the training stage, input images are transformed into the embedding features $\mathbf{f}_i$ by the CNN feature extraction module. Then, the features $\mathbf{f}_i$ are thrown in the fully connect layer (FC) which contains K class center features, and the output $\mathbf{o}_i$ of FC is the similarities of different classes. Later, Hard Decorrelated Centralized Loss is to sort $\mathbf{o}_i$ and select top k of $\mathbf{o}_i$ to calculate softmax loss. After training, the CNN feature extraction module will provide more discriminative representation for fine-grained image retrieval..

$\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$ is as the global centers set $\mathbf{C}$. Inspired by works in image classification, Zheng et al. [7] proposed the Normalize-Scale layer and DGCRL to further improve the performance in FGIR.

$$\mathbf{x}_i = \alpha \mathbf{f}_i, \tag{4}$$

In [7], α is important to prevent the degeneration problem and helps the model to convergence. Readers should refer to [7] for more details.

After obtaining the normalized scale vector $\mathbf{x}_i$, the output of fully connected (FC) layer is

$$\mathbf{o}_i = \mathbf{W}^T \mathbf{x}_i \in R^K, \tag{5}$$

The global loss function DGCRL is proposed to accelerate training stage and is shown as follows

$$L_{\mathrm{DGCRL}} = -\log \frac{\exp^{\mathbf{o}_i(y_i)}}{\sum_j \exp^{\mathbf{o}_i(j)}} + \frac{\lambda}{\|\Omega\|} \sum_{l \neq j} |\mathbf{w}_l \mathbf{w}_j^T|. \tag{6}$$

DGCRL has two parts: the former one is normal softmax loss and the latter one is regularization. λ is the weight value and $\|\Omega\|$ denotes the number of different center pairs. The softmax loss in DGCRL attempts to optimize the intra-class compactness and inter-class separability, and the regularization aims at further enhancing the inter-class separability.

3.2. Hard Decorrelated Centralized Loss

Softmax loss enhances the inner-product $\mathbf{o}_i(y_i) = \mathbf{w}_{y_i}^T \mathbf{x}_i$ belonging to class y_i, and depresses the inner-product $\mathbf{o}_i(j)$ with other classes simultaneously. In other words, softmax loss can depress the similarity with other classes simultaneously. However, in FGIR, it is hard to distinguish similar subspecies as shown in Fig. 1 (emphasized by the orange dotted rectangle). It is important to enhance the ability to distinguishsimilar subspecies. Therefore, we propose Hard Decorrelated Centralized Loss to improve the performance in FGIR. The proposed loss includes two parts: a variant of softmax loss, named Hard Centralized Loss, and the regularization. In this subsection, we will introduce details of our novel loss function.

Firstly, the proposed loss needs to select some similar classes for calculating. Therefore, we use a sort function to obtain a sorted vector as follows

$$\mathrm{Sorted}(\mathbf{o}_i) = \{\mathbf{o}_i(t_1), \mathbf{o}_i(t_2), \ldots, \mathbf{o}_i(t_K)\}, \tag{7}$$

where $\mathbf{o}_i(t_k)$ is the top k value of $\mathbf{o}_i$, and elements are in descending order. Then we select the top $\widehat{K}$ values as denominator of the Hard Centralized Loss (HCL)

$$L_{\mathrm{HCL}} = -\log \frac{\exp^{\mathbf{o}_i(y_i)}}{\sum_{k=1}^{\widehat{K}} \exp^{\mathbf{o}_i(t_k)}}. \tag{8}$$

For simplicity, we define the probability for the j class as:

$$p_j = \frac{\exp^{\mathbf{o}_i(j)}}{\sum_{k=1}^{\widehat{K}} \exp^{\mathbf{o}_i(t_k)}} \in (0, 1). \tag{9}$$

In the stage of back propagation,[1] the gradient of $O_{i,j}$ can be computed as Eq. 10.

[1] In back propagation, the gradient is the negative direction for optimization.

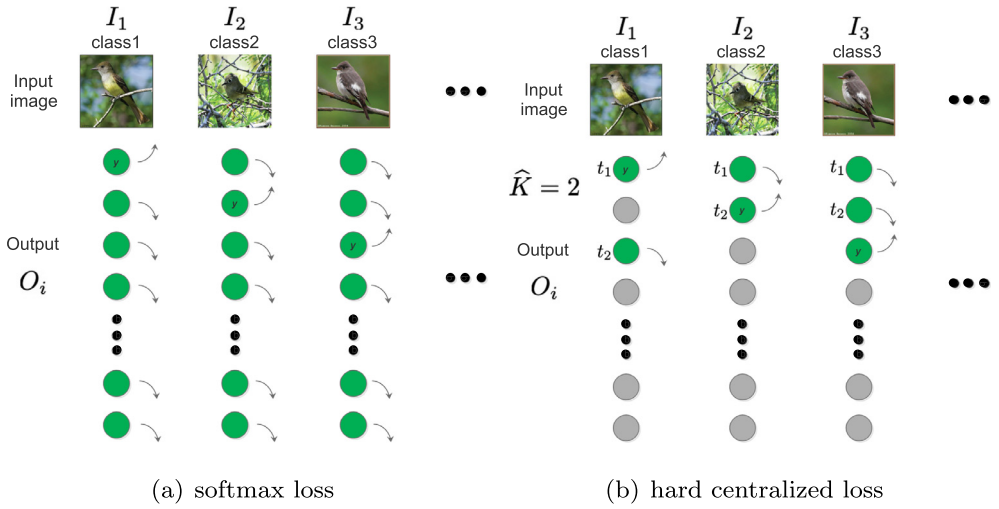


Fig. 4. The pipelines of normal softmax loss and the proposed loss with $\widehat{K} = 2$. These two losses are calculated just with green points of output, and the arrows around output indicate the directions for optimization. Softmax loss attempts to enhance the inner-product O_{i,y_i} belonging to class y_i, and depress the inner-product with other classes simultaneously. Differing from softmax loss, the proposed loss aims at distinguishing the hard similar classes.

$$\frac{\partial L_{\text{HCL}}}{\partial \mathbf{o}_i(j)} = \begin{cases} -1, & j = y_i \text{ and } j \notin \left\{t_1, t_2, \ldots, t_{\widehat{K}}\right\}, \\ -(1-p_j), & j = y_i \text{ and } j \in \left\{t_1, t_2, \ldots, t_{\widehat{K}}\right\}, \\ p_j, & j \neq y_i \text{ and } j \in \left\{t_1, t_2, \ldots, t_{\widehat{K}}\right\}, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

As shows in Eq. 10 and Fig. 4(b), the proposed loss attempts to enhance the inner-product $\mathbf{o}_i(y_i)$ and depress the other top $\widehat{K}$ inner-product $\mathbf{o}_i(t_k), k = 1, 2, \ldots, \widehat{K}$. Different from normal softmax loss, the proposed loss only aims at differentiate the similar classes but neglects dissimilar subspecies.

$$L_{\text{HDCL}} = -\log \frac{\exp^{\mathbf{o}_i(y_i)}}{\sum_{k=1}^{\widehat{K}} \exp^{\mathbf{o}_i(t_k)}} - \frac{\lambda}{\|\Omega\|} \sum_{l \neq j} |\mathbf{w}_l \mathbf{w}_j^T|. \tag{11}$$

Finally, we replace the proposed loss with softmax loss in DGCRL and name it as Hard Decorrelated Centralized Loss (HDCL). HDCL is shown as Eq. 11. The overall framework is summarized in Algorithm 1.

Algorithm 1. Hard Decorrelated Global Centralized Loss.

Input: Training data $\mathscr{I}$; CNN model F;
Input: $\widehat{K}$;
for $t = 1, \ldots, T$epoch do
  Forward image $\mathbf{I}_i$ to feature $\mathbf{f}_i$ by Eq. (1);
  Obtain normalized-scale $\mathbf{x}_i$ by Eq. (4);
  Pass $\mathbf{x}_i$ through FC layer and obtain $\mathbf{o}_i$;
  Sort $\mathbf{o}_i$ and obtain top $\widehat{K}$ value;
  Calculate the loss by Eq. (11);
  Get the gradient of Eq. (11);
Update CNN model F by t^{th} epoch data;
end for
Output: The trained CNN model F

4. Experiments

4.1. Details

Datasets. CUB-200-2011 [1] and Stanford Cars [2] are used as benchmarks, for comparison with state-of-the-art methods. CUB-200-2011 contains 200 subspecies of birds with 11,788 images, and Stanford Cars includes 196 kinds of cars with 16,185 images. They are popular fine-grained datasets in FGIR. Following standard training/testing split setting in previous works [14,7], in CUB-200-2011, we employ the first 100 classes with 5,864 images for training and the rest 100 classes with 5,924 images for testing. In a similar way, for Stanford Cars, we use the first 98 classes with 8,054 images for training, and the rest 98 classes 8,131 images for testing.

Evaluation Protocols. Following the setting of previous works [7], we evaluate the retrieval performance by the standard Recall@**K** [14]. Recall@**K** is the average recall score over all query images in the test set. After extracting the embedding features, we return the top K similar images for each query. In the top K similar images, the recall score will be 1 if it is at least one positive image, and 0 otherwise.

Implementation. We apply ResNet50 [10] as the CNN feature extractor, and it is pretrained on ImageNet ILSVRC-2012 [30] with the Pytorch framework.[2] The extracted features are drawn from the last convolutional layer of ResNet50 with concatenating max-pooling and average-pooling. We train our models via Stochastic Gradient Descent (SGD) with a momentum of 0.9 and weight decay of 5e-6. Besides, we set 0.001 as the initial learning rate, 280 as the crop size, 60 as the mini-batch-size, 100 as the epochs, 0.1 as the weight λ and 100 as the parameter α, respectively. According to extensive trials, we eventually find that the our method achieves advanced performance with these hyper-parameters in a variety of datasets.

Baselines. In our experiments, we compare our method with previous state-of-the-art methods listed as follows. **Contrastive** [13]: training CNN model with pairwise loss. **Triplet** [27]: training CNN model with triplet loss. **LiftedStruct** [14]: training CNN model based on the lifted pairwise distance matrix. **Facility Location** [31]: training CNN model based on a new metric learning scheme for structured prediction, which is aware of the global structure of

[2] https://pytorch.org

Table 1
Recall@**K** on CUB-200-2011 and Stanford Cars with baseline methods. The best results are in **bold**.

Method	CUB-200-2011						Stanford Cars					
Recall@K	1	2	4	8	16	32	1	2	4	8	16	32
Contrastive	26.4	37.7	49.8	62.3	76.4	85.3	21.7	32.3	46.1	58.9	72.2	83.4
Triplet	36.1	48.6	59.3	70.0	80.2	88.4	39.1	50.4	63.3	74.5	84.1	89.8
LiftedStruct	47.2	58.9	70.2	80.2	89.3	93.2	49.0	60.3	72.1	81.5	89.2	92.8
Facility Location	48.2	61.4	71.8	81.9	–	–	58.1	70.6	80.3	87.8	–	–
N-pairs	45.4	58.4	69.5	79.5	–	–	53.9	66.8	77.8	86.4	–	–
Binomial Deviance	52.8	64.4	74.7	83.9	90.4	94.3	–	–	–	–	–	–
Histogram Loss	50.3	61.9	72.6	82.4	88.8	93.7	–	–	–	–	–	–
PDDM+Quadruplet	58.3	69.2	79.0	88.4	93.1	95.7	57.4	68.6	80.1	89.4	92.3	94.9
SCDA	62.2	74.2	83.2	90.1	94.3	97.3	58.5	69.8	79.1	86.2	91.8	95.9
CRL-WSL	65.9	76.5	85.3	90.3	94.4	97.0	63.9	73.7	82.1	89.2	93.7	96.8
DGCRL	67.9	79.1	86.2	91.8	94.8	97.1	75.9	83.9	89.7	94.0	96.6	98.0
HDCL(ours)	**69.5**	**79.6**	**86.8**	**92.4**	**95.6**	**97.5**	**84.4**	**90.1**	**94.1**	**96.5**	**98.0**	**99.0**

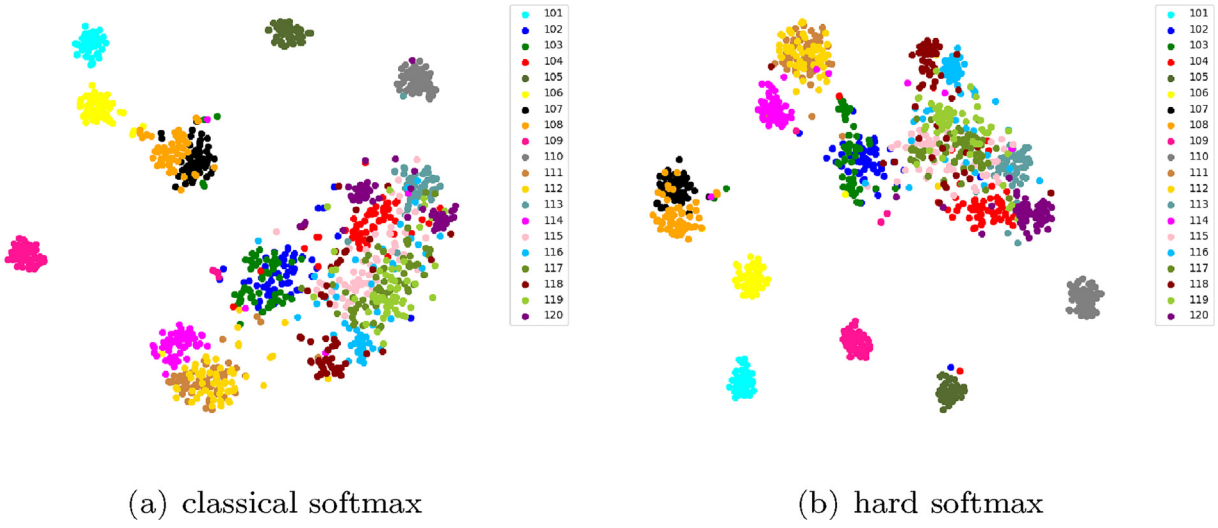


Fig. 5. t-SNE visualizations of the first twenty classes on CUB-200-2011.

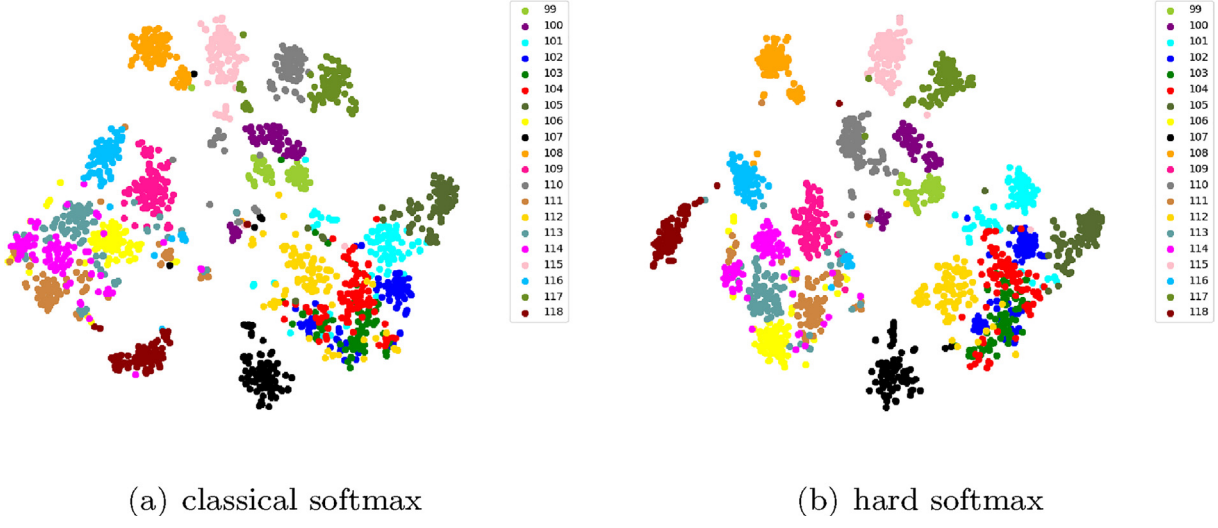
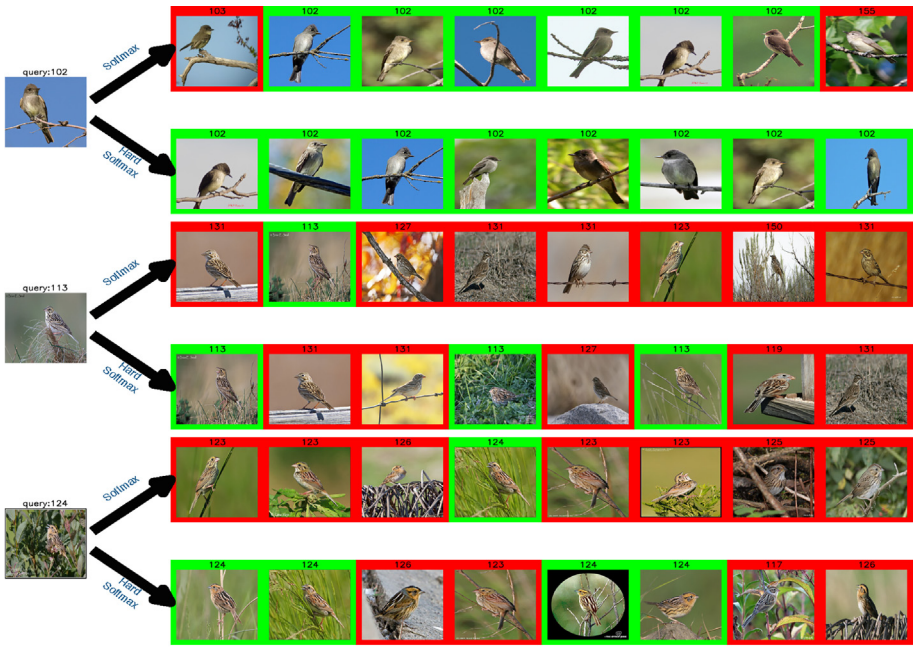
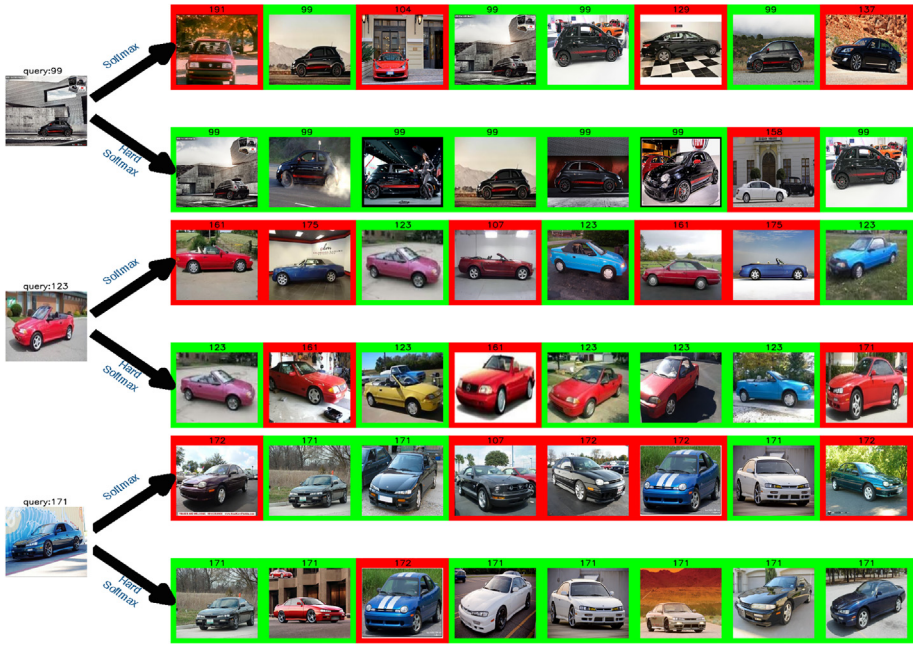


Fig. 6. t-SNE visualizations of the first twenty classes on Stanford Cars.

the embedding space. **N-pairs** [24]: employing multi-class N-pair loss to address the slow convergence from pairwise and triplet loss. **Binomial Deviance** [25]: evaluates the cost between similarities, which is proven to be outliers. **Histogram Loss** [25]: penalizing the overlap between distributions of positive pairs' distances and distributions of negative pairs' distance. **PDDM+Quadruplet** [26]: choosing hard positive examples and negative examples to train CNN and adopting PDDM block to evaluate the similarities. **SCDA**



(a) CUB-200-2011



(b) Stanford Cars

Fig. 7. Some results of fine-grained image retrieval on two benchmarks with different methods(classical softmax loss VS hard softmax loss). From left to right, the first part of image is the category of query. The second part is the results of the top 8 similar images of query in the test set. When the retrieved image matches the category of the query image, it will be highlighted in a green box, otherwise, it will be highlighted in red box.

[5]: selecting discriminative and representative examples in the last convolution layer of VGG16 without fine-tuning, and then combining max-pooling features and average-pooling features. **CRL-WSL** [6]: training CNN model with a centralized ranking loss, and using weakly supervised localization method to obtain features under pixel-levelobject localization. **DGCRL** [7]: adding the Normalized-Scale layer into the ResNet-50 based model and using DGCRL (a global structure loss) to train the model.

4.2. Performance

Table 1 shows that our models achieve state-of-the-art methods in terms of Recall@K in these two benchmarks. The ResNet50 model trained with HDCL obtains 69.5 Recall@1 in CUB-200-2011 and 84.4 Recall@1 in Stanford Cars, respectively. Comparing with previous work [7], the main difference is deploying different losses in the training period, which suggests that the proposed loss can

Table 2
Retrieval details of six fine-grained image datasets.

Dataset	Object	Training		Testing	
		images	classes	images	classes
CUB-200-2011 [1]	Bird	5864	100	5924	100
Stanford Cars [2]	Car	8054	98	8131	98
Stanford Dogs [3]	Dog	10651	60	9929	60
FGVC aircraft [32]	Aircraft	5000	50	5000	50
Oxford Flowers [33]	Flower	3493	60	4696	60
Oxford Pets [34]	Pet	3766	19	3583	18

Table 3
Retrieval results of VGG with different methods. The best results are in **bold**.

Setting			Evaluation											
Backbone	Dataset	Method	Recall@K						Precision@K			mAP@K		
			1	2	4	8	16	32	1	5	10	1	5	10
VGG19	Cars	SCDA	49.7	62.0	72.3	81.4	89.4	94.9	49.7	35.8	29.5	49.7	56.4	52.6
		CRL	70.6	80.3	87.1	92.1	95.6	97.8	70.6	57.2	49.1	70.6	74.3	69.5
		DGCRL	81.6	87.8	92.4	95.5	97.4	98.7	81.6	71.3	63.8	81.6	83.8	79.8
		HDCL	**82.9**	**89.3**	**93.4**	**96.3**	**97.9**	**99.0**	**82.9**	**72.1**	**65.2**	**82.9**	**84.5**	**80.0**
	CUB	SCDA	57.6	69.1	79.7	87.9	93.3	96.7	57.6	49.5	44.4	57.6	64.2	60.5
		CRL	61.0	72.3	81.4	88.3	93.4	96.4	61.0	54.0	49.5	61.0	67.0	63.7
		DGCRL	61.6	72.8	81.0	87.8	92.7	96.0	61.6	54.6	49.3	61.6	67.4	64.3
		HDCL	**64.3**	**74.3**	**83.3**	**89.6**	**94.3**	**96.8**	**64.3**	**57.3**	**52.8**	**64.3**	**69.6**	**66.2**
	Dogs	SCDA	82.5	90.0	94.5	97.1	98.4	99.1	82.5	79.4	77.4	82.5	86.1	83.9
		CRL	82.5	90.0	**94.2**	**96.8**	98.2	**99.0**	82.5	**79.6**	**77.7**	82.5	86.0	84.0
		DGCRL	82.5	89.5	94.1	96.7	98.1	98.9	82.5	**79.6**	**77.7**	82.6	85.9	84.0
		HDCL	**82.9**	**90.1**	94.1	96.7	**98.3**	**99.0**	**82.9**	**79.6**	**77.7**	**82.9**	**86.1**	**84.1**
	FGVC	SCDA	61.3	71.5	80.0	87.9	93.1	96.4	61.3	49.3	42.5	61.3	66.4	62.4
		CRL	68.1	78.5	86.6	91.8	95.4	93.4	68.1	57.1	51.0	68.1	72.7	68.0
		DGCRL	70.5	80.0	87.8	92.7	95.3	97.4	70.5	61.1	55.5	70.5	75.0	70.8
		HDCL	**73.1**	**82.1**	**88.8**	**93.2**	**96.0**	**97.6**	**73.1**	**63.3**	**57.7**	**73.1**	**77.0**	**72.6**
	Flowers	SCDA	86.6	91.3	94.9	97.4	98.7	99.3	86.6	79.9	74.8	86.6	88.3	85.5
		CRL	91.3	95	97.1	98.5	99.2	99.6	91.3	85.8	81.4	91.3	92.2	89.8
		DGCRL	91.9	95.5	97.3	98.5	**99.4**	**99.8**	91.9	87.2	82.9	91.9	93.0	90.7
		HDCL	**92.8**	**95.8**	**97.7**	**98.8**	99.3	99.7	**92.8**	**87.7**	**83.8**	**92.8**	**93.5**	**91.3**
	Pets	SCDA	96.6	**98.5**	**99.3**	**99.7**	**99.8**	**99.9**	96.6	95.0	94.1	96.6	97.1	96.3
		CRL	96.6	98.4	99.1	99.5	**99.8**	99.8	96.6	95.3	94.5	96.6	97.2	96.4
		DGCRL	96.6	98.1	99.0	99.5	**99.8**	**99.9**	96.6	**95.4**	**94.6**	96.6	97.2	**96.5**
		HDCL	**96.9**	98.4	99.2	99.6	99.7	**99.9**	**96.9**	**95.4**	94.4	**96.9**	**97.3**	**96.5**

significantly improve the performance in FGIR. In the following, we intend to further clarify the effectiveness of the developed loss from the perspective of visualization. On the one hand, as the tsne visualizations of the first twenty classes on two prevalent datasets that are shown in Fig. 5 and Fig. 6, we find that the model trained via the hard softmax loss shows favorable intra-class compactness and inter-class separability compared with the state-of-the-art method [7] that via classical softmax. On the other hand, we give the results of image retrieval on CUB-200-2011 and Stanford Cars with different methods respectively in Fig. 7a) and Fig. 7(b). It is natural to investigate that the model returns similar images of the same class despite variations in view point, pose and background, which, again proves that the novel loss has the ablity to drie extremely similar categories (hard negative) to separate automatically.

5. Discussion

5.1. Extra Experiments for Fine-Grained Image Retrieval

For further exploring the effect of the proposed loss function, we have carried on another four experiments on different fine-grained dataset: Stanford Dogs [3], FGVC aircraft [32], Oxford Flowers [33] and Oxford Pets [34]. Following the setting of FGIR, the details of six datasets for retrieval is shown on Table 2.

In the experiments, we compared the performance of the combination of different losses (e.g. SCDA, CRL, DGCRL and HDCL) and different backbones (e.g. VGG19 [9], ResNet50 and ResNet101 [10]) for general retrieval task.[3] We selected Recall@K, Precision@K and mAP@K as the evaluations, for thoroughly reflecting the retrieval performance of methods. Table 3, Table 4 and Table 5 illustrate the results of VGG19, ResNet50 and ResNet101, respectively.

Tables 3–5 show that models trained by the proposed loss function usually obtain the best retrieval performances in the six given benchmarks, thus it prove that our novel loss is beneficial for FGIR. It is worth noting that the benefits brought by the new loss are not the same on different datasets. We find that our loss function is hard to attain an ideal improvement on Oxford Flowers [33] and Oxford Pets [34], probably because the inter-class similarity of them are less than other datasets.

5.2. Ablation Study of $\widehat{K}$

There are three hyperparameters in our experiments: the hyperparameter $\widehat{K}$ in the proposed loss, the scale value α, and the weight λ. In this section, we discuss the effect of our proposed loss in FGIR, and set $\widehat{K}$ as different values ($\widehat{K} = K, 10, 5, 2$). Therefore, in our experiments, we freeze α as 100 and λ as 0.1. The proposed loss with $\widehat{K} = K$ is the same as the normal softmax loss

[3] All the experiments are based on our reimplements within the same machine.

Table 4
Retrieval results of ResNet50 with different methods. The best results are in **bold**.

Setting			Evaluation											
Backbone	Dataset	Method	Recall@K						Precision@K			mAP@K		
			1	2	4	8	16	32	1	5	10	1	5	10
ResNet50	Cars	SCDA	48.3	60.2	71.8	81.8	90.2	95.7	48.3	35.8	29.4	48.3	55.4	51.9
		CRL	57.8	69.1	78.6	86.6	92.4	96.3	57.8	43.5	36.3	57.8	63.4	59
		DGCRL	82.3	88.1	92.5	95.3	97.6	98.7	82.3	74.4	67.9	82.3	85.5	81.6
		HDCL	**84.4**	**90.1**	**94.1**	**96.5**	**98.0**	**99.0**	**84.4**	**75.4**	**68.8**	**84.4**	**86.2**	**82.5**
	CUB	SCDA	57.3	70.2	81.0	88.4	93.9	96.9	57.3	48.1	43.0	57.3	64.2	60.2
		CRL	62.5	74.2	82.9	89.7	94.3	96.9	62.5	54.8	49.6	62.5	68.6	64.8
		DGCRL	67.1	77.6	85.6	91.0	95.2	96.8	67.1	61.8	55.3	67.1	72.4	69.7
		HDCL	**69.5**	**79.6**	**86.8**	**92.4**	**95.6**	**97.5**	**69.5**	**62.9**	**58.2**	**69.5**	**74.3**	**71.0**
	Dogs	SCDA	87.1	**93.1**	**96.3**	**98.0**	**99.1**	**99.5**	87.1	85.0	83.6	87.1	**89.9**	88.3
		CRL	85.8	91.9	95.8	97.9	98.8	99.4	85.8	83.5	82.0	85.8	88.8	87.0
		DGCRL	87.3	92.7	95.7	97.7	98.7	99.2	87.3	85.2	83.8	87.3	89.8	88.3
		HDCL	**88.1**	92.8	95.9	97.6	98.7	99.3	**88.1**	**85.3**	**84.2**	**88.1**	**89.9**	**88.5**
	FGVC	SCDA	56.5	67.7	77.6	85.7	92.0	96.2	56.5	45.4	39.2	56.5	62.7	58.7
		CRL	61.1	71.6	80.9	88.2	93.1	96.3	61.1	50.1	43.7	61.6	66.5	62.3
		DGCRL	70.1	79.6	88.0	93.0	95.8	97.6	70.1	60.3	54.2	70.1	74.5	70.3
		HDCL	**71.1**	**81.0**	**88.3**	**93.3**	**96.3**	**98.0**	**71.1**	**61.2**	**55.2**	**71.1**	**75.3**	**71.0**
	Flowers	SCDA	90.7	94.3	97.0	98.4	99.1	99.7	90.7	84.7	80.3	90.7	91.7	89.1
		CRL	94.3	96.7	98.3	99.0	99.5	99.7	94.3	89.7	86.0	94.3	94.8	92.8
		DGCRL	95.4	97.7	98.7	**99.4**	99.7	**99.9**	95.4	92.3	89.6	95.4	96.0	94.5
		HDCL	**95.9**	**98.1**	**98.9**	**99.4**	**99.8**	**99.9**	**95.9**	**92.5**	**89.7**	**95.9**	**96.3**	**94.8**
	Pets	SCDA	98.2	99.0	99.5	99.7	**99.9**	**99.9**	98.2	97.0	96.3	98.2	98.2	97.7
		CRL	98.0	99.1	99.6	99.7	**99.9**	**99.9**	98.0	97.1	96.5	98.0	98.3	97.9
		DGCRL	98.4	99.3	99.6	99.7	99.8	**99.9**	98.4	97.5	97.1	98.4	98.7	98.2
		HDCL	**98.8**	**99.4**	**99.8**	**99.9**	**99.9**	**99.9**	**98.8**	**97.6**	**97.2**	**98.8**	**98.8**	**98.4**

Table 5
Retrieval results of ResNet101 with different methods. The best results are in **bold**.

Setting			Evaluation											
Backbone	Dataset	Method	Recall@K						Precision@K			mAP@K		
			1	2	4	8	16	32	1	5	10	1	5	10
ResNet101	Cars	SCDA	52.9	65.2	76.4	85.3	92.1	96.5	52.9	39.9	33.4	52.9	59.7	55.5
		CRL	75.8	84.1	90.4	94.7	97.0	98.8	75.8	63.9	56.5	75.8	78.8	74.2
		DGCRL	83.0	89.5	93.2	95.5	97.4	98.7	83.0	74.9	67.8	83.0	85.1	58.1
		HDCL	**85.1**	**90.8**	**94.5**	**97.0**	**98.4**	**99.2**	**85.1**	**75.7**	**69.1**	**85.1**	**86.4**	**59.3**
	CUB	SCDA	62.3	73.8	83.1	90.0	94.3	96.9	62.3	53.7	48.7	62.3	68.4	64.3
		CRL	69.3	79.1	86.6	92.2	95.2	97.2	69.3	62.2	57.4	69.3	74.0	70.5
		DGCRL	69.5	79.7	**87.3**	92.1	95.3	97.3	69.5	63.5	59.0	69.5	74.8	71.3
		HDCL	**70.9**	**80.4**	**87.3**	**92.6**	**95.6**	**97.4**	**70.9**	**64.6**	**60.4**	**70.9**	**75.5**	**72.3**
	Dogs	SCDA	89.0	94.1	**97.1**	**98.3**	**99.0**	**99.5**	89.0	87.5	86.5	89.0	91.4	90.1
		CRL	**89.4**	**94.2**	96.9	98.2	98.9	99.4	**89.4**	**87.7**	**86.7**	**89.4**	**91.6**	**90.3**
		DGCRL	88.5	93.4	96.3	97.9	98.7	99.2	88.5	86.8	85.7	88.5	90.9	89.6
		HDCL	**89.4**	93.9	96.6	98.0	98.8	99.3	**89.4**	87.4	86.4	**89.4**	91.4	90.1
	FGVC	SCDA	58.4	69.4	78.7	87.1	93.0	96.5	58.4	46.3	39.7	58.4	64.2	60.3
		CRL	66.6	77.5	86.3	91.9	95.3	97.7	66.6	55.1	49.0	66.6	71.4	67.0
		DGCRL	67.5	78.3	86.7	92.4	95.4	97.6	67.5	57.9	52.2	67.5	72.6	68.2
		HDCL	**69.7**	**80.2**	**88.0**	**93.6**	**96.4**	**97.9**	**69.7**	**59.1**	**53.0**	**69.7**	**74.2**	**69.7**
	Flowers	SCDA	90.7	94.4	96.9	98.0	99.1	**99.8**	90.7	84.7	80.2	90.7	91.7	89.1
		CRL	94.7	96.9	98.1	99.0	**99.5**	99.7	94.7	90.5	87.2	94.7	95.1	93.2
		DGCRL	94.9	97.0	98.2	99.0	99.4	99.6	94.9	90.7	87.4	94.9	95.2	93.4
		HDCL	**95.7**	**97.5**	**98.5**	**99.2**	**99.5**	99.7	**95.7**	**92.2**	**88.9**	**95.7**	**95.9**	**94.4**
	Pets	SCDA	97.9	98.9	**99.6**	**99.8**	**99.9**	**99.9**	97.9	97.1	96.5	97.9	98.2	97.8
		CRL	**98.2**	**99.1**	**99.6**	99.7	**99.9**	**99.9**	**98.2**	97.3	**96.9**	**98.2**	**98.4**	**98.0**
		DGCRL	98.1	98.9	99.4	99.7	**99.9**	**99.9**	98.1	97.3	96.8	98.1	98.3	**98.0**
		HDCL	**98.2**	98.9	99.4	99.7	**99.9**	**99.9**	**98.2**	**97.4**	**96.9**	**98.2**	**98.4**	**98.0**

function. The proposed loss with $\widehat{K} = 10$ means the training images need to be distinguished into the top 10 similar classes. Along this line, $\widehat{K} = 5, 2$ means the training images need to be distinguished into the top 5, 2 similar classes, respectively. We can find that when the loss is normal softmax loss, the performance of Stanford Cars is better than [7] but CUB-200-2011 is worse. It means that the hyperparameters of our experiment are different from previous work in [7], and they are much suitable for the Stanford Cars dataset.

Furthermore, we employ the proposed loss function to train different CNN models. Table 6 shows results on CUB-200-2011 [1] and Stanford Cars [2] of different $\widehat{K}$ with VGG19, ResNet50 and ResNet101. We can find that our proposed loss function is beneficial for improving the performance of FGIR. Furthermore, when $\widehat{K} = 2$, ResNet50, and ResNet101 achieve best Recall@1 performance in these two benchmarks. In summary, it means that the most important thing in FGIR is to separate the hard negative class.

Table 6
Retrieval results of HDCL with different $\widehat{K}$ on CUB-200-2011 and Stanford Cars. The best results are in **bold**.

Setting			Evaluation											
Dataset	Backbone	$\widehat{K}$	Recall@K						Precision@K			mAP@K		
			1	2	4	8	16	32	1	5	10	1	5	10
CUB	VGG19	100 (CE)	61.6	72.8	81.0	87.8	92.7	96.0	61.6	54.6	49.3	61.6	67.4	64.3
		10	63.0	73.2	82.2	88.9	93.5	96.2	63.0	56.0	51.4	63.0	68.3	65.1
		5	**64.3**	74.2	82.7	88.6	93.4	96.6	**64.3**	57.2	52.3	64.3	69.5	66.0
		2	**64.3**	**74.3**	**83.3**	**89.6**	**94.3**	**96.8**	**64.3**	**57.3**	**52.8**	**64.3**	**69.6**	**66.2**
	ResNet50	100 (CE)	67.1	77.6	85.6	91.0	95.2	96.8	67.1	61.8	55.3	67.1	72.4	69.7
		10	69.0	78.8	**86.9**	92.0	**95.6**	**97.5**	69.0	62.7	**58.3**	69.0	74.0	70.7
		5	69.1	78.9	86.2	92.1	95.5	97.4	69.1	62.2	57.6	69.1	74.0	70.6
		2	**69.5**	**79.6**	86.8	**92.4**	**95.6**	**97.5**	**69.5**	**62.9**	58.2	**69.5**	**74.3**	**71.0**
	ResNet101	100 (CE)	69.5	79.7	**87.3**	92.1	95.3	97.3	69.5	63.5	59.0	69.5	74.8	71.3
		10	70.3	80.1	87.15	92.7	**96.0**	**97.7**	70.3	64.5	60.2	70.3	74.9	71.9
		5	70.5	**80.5**	**87.7**	**92.8**	95.7	97.6	70.5	64.4	60.1	70.5	75.2	72.1
		2	**70.9**	80.4	87.3	92.6	95.6	97.4	**70.9**	**64.6**	**60.4**	**70.9**	**75.5**	**72.3**
Cars	VGG19	98 (CE)	81.6	87.8	92.4	95.5	97.4	98.7	81.6	71.3	63.8	81.6	83.8	79.8
		10	81.9	88.3	92.6	95.7	97.7	98.7	81.9	71.8	64.6	81.9	83.9	79.9
		5	82.1	88.8	93.0	95.8	97.5	98.8	82.1	71.5	64.1	82.1	84.2	79.8
		2	**82.9**	**89.3**	**93.4**	**96.3**	**97.9**	**99.0**	**82.9**	**72.1**	**65.2**	**82.9**	**84.5**	**80.0**
	ResNet50	98 (CE)	82.3	88.1	92.5	95.3	97.6	98.7	82.3	74.4	67.9	82.3	85.5	81.6
		10	84.3	90.3	94.2	**96.7**	**98.2**	99.0	84.3	74.9	68.1	84.3	86.0	82.2
		5	84.3	**90.4**	**94.3**	96.6	**98.2**	**99.1**	84.3	75.2	68.6	84.3	**86.2**	82.3
		2	**84.4**	90.1	94.1	96.5	98.0	99.0	**84.4**	**75.4**	**68.8**	**84.4**	**86.2**	**82.5**
	ResNet101	98 (CE)	83.0	89.5	93.2	95.5	97.4	98.7	83.0	74.9	67.8	83.0	85.1	58.1
		10	84.8	90.4	94.4	96.8	98.2	99.1	84.8	75.2	68.9	84.8	86.2	**59.3**
		5	**85.1**	90.6	**94.6**	96.9	98.3	**99.2**	**85.1**	75.4	68.9	**85.1**	86.3	**59.3**
		2	**85.1**	**90.8**	94.5	**97.0**	**98.4**	**99.2**	**85.1**	**75.7**	**69.1**	**85.1**	**86.4**	**59.3**

Table 7
The recognition performances of different loss function.

Model	Loss	Cars	CUB	Dogs	FGVC	Flowers	Pets
VGG19 [9]	Softmax loss	83.2%	77.2%	82.6%	79.1%	**90.0%**	92%
	Hard Softmax loss	**83.5%**	**77.3%**	**83.0%**	**79.6%**	89.9%	**92.3%**
ResNet50 [10]	Softmax loss	87.3%	80.3%	87.7%	**82.1%**	95.2%	93.7%
	Hard Softmax loss	**88.2%**	**82.1%**	**88.6%**	78.7%	**95.5%**	**93.9%**
Densenet161 [29]	Softmax loss	89.0%	83.6%	88.0%	**84.2** %	96.6%	94.1 %
	Hard Softmax loss	**89.2%**	**83.8%**	**89.1%**	82.2%	**97.2%**	**94.1%**
InceptionV3 [36]	Softmax loss	**86.0%**	79.2%	90.5%	**81.5%**	**93.7%**	**93.7%**
	Hard Softmax loss	85.7%	**80.1%**	**91.0%**	79.8%	93.2%	93.4%

5.3. Effect in Fine-grained object recognition

We also consider the influence of the proposed loss in Fine-grained object recognition [35], because the proposed Hard Centralized Loss is a variant of Softmax loss. Four different CNN models are trained with different loss functions (Softmax loss and HCL) on six fine-grained recognition datasets.[4]

These models are trained by the SGD optimizer, with the mini-batch-size of 32, the initial learning rate of 0.001, the momentum of 0.9, training epochs of 400, and the weight decay of 5e-4. As summarized in Table 7, the proposed loss HCL usually be helpful to promote the performance of FGOR. However, according to the observation in experiments, sometimes training with HCL is unstable in the beginning on some datasets, e.g. FGVC dataset. We consider that the main reason for such instability may be the initial classifier cannot find the correct negative centers for training CNN models. Therefore, we suggest that one should use the original softmax loss to train the models in the first few epochs, and then switch to HCL later.

6. Conclusion

Due to the small diversity in inter-class but large diversity within the intra-class, the tasks of fine-grained image analysis are challenging. And it is hard to separate the similar classes in FGIR. In this paper, we propose a Hard Centralized Loss function to enforce the CNN model to separate similar classes. In the training stage, the proposed loss function attempts to enhance the inner-product belonging to the label and depress the inner-product of other $\widehat{K}$ similar classes. In our experiments, we find that the proposed loss function can improve the performance in FGIR, and achieve state-of-the-art performances on two fine-grained retrieval benchmarks. Furthermore, the proposed loss function can also commonly improve the performance of fine-grained object recognition. Considering to fuse local structure loss and global structure loss to improve the fine-grained image tasks can be our future work [37,38].

CRediT authorship contribution statement

Xianxian Zeng: Conceptualization, Methodology, Investigation, Software, Data curation, Writing - original draft. **Shun Liu:** Writing - review & editing, Formal analysis, Validation. **Yun Zhang:** Conceptualization, Methodology, Supervision, Project administration, Funding acquisition. **Xiaodong Wang:** Software, Data curation. **Kairui Chen:** Writing - review & editing. **Dong Li:** Writing - review & editing, Supervision.

[4] In our experiments, the CNN models are not the best performances in these datasets, because our experiments are based on one RTX2080ti GPU and the hyperparameters may be different from their previous works.

Declaration of Competing Interest

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

Acknowledgments

This work was supported by National Natural Science Foundation of China: U1501251, 61503084 and Natural Science Foundation of Guangdong Province, China: 2021A1515011867.

References

[1] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200-2011 dataset..
[2] J. Krause, H. Jin, J. Yang, L. Fei-Fei, Fine-grained recognition without part annotations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5546–5555.
[3] A. Khosla, N. Jayadevaprakash, B. Yao, F.-F. Li, Novel dataset for fine-grained image categorization: Stanford dogs, in: Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC), Vol. 2, 2011..
[4] L. Xie, J. Wang, B. Zhang, Q. Tian, Fine-grained image search, IEEE Trans. Multimedia 17 (5) (2015) 636–647.
[5] X.-S. Wei, J.-H. Luo, J. Wu, Z.-H. Zhou, Selective convolutional descriptor aggregation for fine-grained image retrieval, IEEE Trans. Image Process. 26 (6) (2017) 2868–2881.
[6] X. Zheng, R. Ji, X. Sun, Y. Wu, F. Huang, Y. Yang, Centralized ranking loss with weakly supervised localization for fine-grained object retrieval, IJCAI (2018) 1226–1233.
[7] X. Zheng, R. Ji, X. Sun, B. Zhang, Y. Wu, F. Huang, Towards optimal fine grained retrieval via decorrelated centralized loss with normalize-scale layer..
[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105..
[9] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the IEEE International Conference on Learning Representations, 2015.
[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[11] D. Lopez-Sanchez, A. Gonzalez Arrieta, J. M. Corchado, Visual content-based web page categorization with deep transfer learning and metric learning, Neurocomputing 338 (APR.21) 418–431..
[12] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149.
[13] S. Bell, K. Bala, Learning visual similarity for product design with convolutional neural networks, ACM Transactions on Graphics (TOG) 34 (4) (2015) 98..
[14] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4004–4012.
[15] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, Y. Wu, Learning fine-grained image similarity with deep ranking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1386–1393.
[16] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.
[17] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3384–3391.
[18] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE transactions on pattern analysis and machine intelligence 34 (9) (2011) 1704–1716.
[19] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, in: European conference on computer vision, Springer, 2014, pp. 392–407.
[20] A. Babenko, V. Lempitsky, Aggregating deep convolutional features for image retrieval, arXiv preprint arXiv:1510.07493..
[21] G. Tolias, R. Sicre, H. Jégou, Particular object retrieval with integral max-pooling of cnn activations, arXiv preprint arXiv:1511.05879..
[22] H. Noh, A. Araujo, J. Sim, T. Weyand, B. Han, Large-scale image retrieval with attentive deep local features, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3456–3465.
[23] F. Radenović, G. Tolias, O. Chum, Fine-tuning cnn image retrieval with no human annotation, IEEE transactions on pattern analysis and machine intelligence 41 (7) (2018) 1655–1668.
[24] K. Sohn, Improved deep metric learning with multi-class n-pair loss objective, in: Advances in Neural Information Processing Systems, 2016, pp. 1857–1865..
[25] E. Ustinova, V. Lempitsky, Learning deep embeddings with histogram loss, in: Advances in Neural Information Processing Systems, 2016, pp. 4170–4178..
[26] C. Huang, C. C. Loy, X. Tang, Local similarity-aware deep feature embedding, in: Advances in neural information processing systems, 2016, pp. 1262–1270..
[27] X. Zhang, F. Zhou, Y. Lin, S. Zhang, Embedding label structures for fine-grained feature representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1114–1123.
[28] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.
[29] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, IEEE, 2009, pp. 248–255..
[31] H. Oh Song, S. Jegelka, V. Rathod, K. Murphy, Deep metric learning via facility location, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5382–5390.
[32] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*..
[33] M. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 2, 2006, pp. 1447–1454. https://doi.org/10.1109/CVPR.2006.42..
[34] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. V. Jawahar, Cats and dogs, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3498–3505. https://doi.org/10.1109/CVPR.2012.6248092..
[35] J. Zhao, Y. Peng, X. He, Attribute hierarchy based multi-task learning for fine-grained image classification, Neurocomputing 395 (2020) 150–159, https://doi.org/10.1016/j.neucom.2018.02.109. http://www.sciencedirect.com/science/article/pii/S0925231219308938.
[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
[37] J. Yu, Y. Yang, F. Murtagh, X. Gao, Fine-grained visual understanding and reasoning, Neurocomputing 398 (2020) 408–410, https://doi.org/10.1016/j.neucom.2019.07.055.
[38] Y. Yan, B. Ni, H. Wei, X. Yang, Fine-grained image analysis via progressive feature learning, Neurocomputing 396 (2020) 254–265, https://doi.org/10.1016/j.neucom.2018.07.100. http://www.sciencedirect.com/science/article/pii/S0925231219304400.

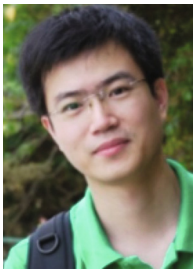
Xianxian Zeng was born in Guangzhou, Guangdong, China in 1992. He received the bachelor's degree and PhD degree in the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2015 and 2020, respectively. He is currently as a lecturer in the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China. His research interestes include visual recognition, deep learning and 3D reconstruction.

Shun Liu received the B.Eng. Degree and Ph.D. from the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2015 and 2020, respectively. He is currently as a lecturer in the School of Automation, Guangdong Polytechnic Normal University, Guangzhou, China. His research interests include statistical signal processing, pattern recognition, and machine learning.

Yun Zhang received the B.S. and M.S. degrees in automatic engineering from Hunan University, Changsha, China, in 1982 and 1986, respectively, and the Ph.D. degree in automatic engineering from the South China University of Science and Technology, Guangzhou, China, in 1998. He is currently a full Professor with the School of Automation, Guangdong University of Technology, Guangzhou, China.

Xiaodong Wang received the bachelor's degree in the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2016. He is currently studying for a PhD. degree in the Department of Automation, GuangDong University of Technology, Guangzhou, China.

Kairui Chen received the Ph. D degree in the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2017. From December 2015 to December 2016, he was a Visiting Scholar with the Automation and Robotics Research Institute, University of Texas at Arlington. He was a Postdoctor with the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2019. Currently, He is an associate professor with Guangzhou University. His research interests include multi-agent system control, neural networks learning, adaptive control and optimal control.



Dong Li is an associate professor in the faculty of Automation at Guangdong University of Technology (GDUT). He received his PhD degree in the Department of Electronic and Information Engineering at the Hong Kong Polytechnic University in Feb 2014. He worked under the supervision of Professor Kin-Man (Kenneth) Lam. His research interest lies in computer vision, pattern recognition and image analysis. His earlier work designed robust and distinctive features to describe pore-scale facial keypoints, such as pores, fine winkles and hair. More specifically, he focus on: 1) designing new pore-scale feature extraction algorithms, 2) developing pore-scale facial feature applications, such as face verification, 3) adapting existing algorithms to pore-scale application. In the area of image analysis, he is especially interested in color correction and image restoration.**Dong Li** is an associate professor in the faculty of Automation at Guangdong University of Technology (GDUT). He received his PhD degree in the Department of Electronic and Information Engineering at the Hong Kong Polytechnic University in Feb 2014. He worked under the supervision of Professor Kin-Man (Kenneth) Lam. His research interest lies in computer vision, pattern recognition and image analysis. His earlier work designed robust and distinctive features to describe pore-scale facial keypoints, such as pores, fine winkles and hair. More specifically, he focus on: 1) designing new pore-scale feature extraction algorithms, 2) developing pore-scale facial feature applications, such as face verification, 3) adapting existing algorithms to pore-scale application. In the area of image analysis, he is especially interested in color correction and image restoration.