Fine-Grained Image Classification Using Modified DCNNs Trained by Cascaded Softmax and Generalized Large-Margin Losses

Weiwei Shi, Yihong Gong, *Fellow, IEEE*, Xiaoyu Tao, De Cheng, and Nanning Zheng, *Fellow, IEEE*

***Abstract*— We develop a fine-grained image classifier using a general deep convolutional neural network (DCNN). We improve the fine-grained image classification accuracy of a DCNN model from the following two aspects. First, to better model the *h*-level hierarchical label structure of the fine-grained image classes contained in the given training data set, we introduce *h* fully connected (fc) layers to replace the top fc layer of a given DCNN model and train them with the cascaded softmax loss. Second, we propose a novel loss function, namely, generalized large-margin (GLM) loss, to make the given DCNN model explicitly explore the hierarchical label structure and the similarity regularities of the fine-grained image classes. The GLM loss explicitly not only reduces between-class similarity and within-class variance of the learned features by DCNN models but also makes the subclasses belonging to the same coarse class be more similar to each other than those belonging to different coarse classes in the feature space. Moreover, the proposed fine-grained image classification framework is independent and can be applied to any DCNN structures. Comprehensive experimental evaluations of several general DCNN models (AlexNet, GoogLeNet, and VGG) using three benchmark data sets (Stanford car, fine-grained visual classification-aircraft, and CUB-200-2011) for the fine-grained image classification task demonstrate the effectiveness of our method.**

***Index Terms*— Cascaded softmax loss, deep convolutional neural network (DCNN), fine-grained image classification, generalized large-margin (GLM) loss, hierarchical label structure.**

I. Introduction

FINE-GRAINED image classification aims to recognize subordinate classes of some base class, such as different models of cars [1]–[5], species of birds [5]–[9], variants of aircrafts [10], [11], and so on. It has a wide range of applications, such as vehicle model recognition for video surveillance, fine-grained image content annotation, vertical search, and so on. The challenges of fine-grained image classification mainly come from the following two aspects: between-class similarity and within-class variance [12]–[16]. On one hand, visual differences between different fine-grained classes could be very small and subtle. On the other hand, instances belonging to the same fine-grained class may have significantly varied appearances due to different locations, viewpoints, poses, lighting conditions, and so on. For example, the "California gulls" shown in Fig. 1(a) are visually quite similar to the "ringed-beak gulls" in Fig. 1(b), and the only salient difference between them lies in the patterns on their beaks. Meanwhile, caused by differences in pose, viewpoint, and lighting condition, the "California gulls" in the different images (so as the "ringed-beak gulls") exhibit remarkably different appearances between each other.

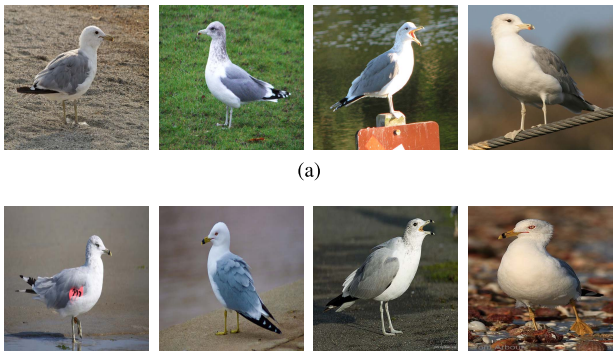


Fig. 1. Sample images of (a) California gull and (b) ringed-beak gull. The salient difference between them lies in the patterns on their beaks.

For the fine-grained image classification task, many part-based methods [17]–[20] have been proposed in the literature. These methods first detect different parts of the target object and, then, model the local parts' appearances to increase the between-class discrimination and reduce the within-class variance at the same time. For example, for fine-grained birds classification, Zhang *et al.* [18] proposed to learn the appearance models of such parts as head, beak, and body, and to enforce the geometric constraints between them. However, part-based methods rely on accurate parts detection, which is another challenging problem that may fail in the presence of occlusions and large viewpoint/pose variations. Moreover, the parts' detectors are usually trained in a supervised manner that requires a sufficient amount of training samples. Obviously, annotating object parts is significantly more challenging and expensive than assigning fine-grained image labels.

Manuscript received October 3, 2017; revised April 9, 2018; accepted June 27, 2018. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2015CB351705 and in part by the National Natural Science Foundation of China under Grant 61332018. *(Corresponding author: Yihong Gong.)*

W. Shi, Y. Gong, X. Tao, and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China, and also with the National Engineering Laboratory for Visual Information Processing and Applications, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: shiweiwei.math@stu.xjtu.edu.cn; ygong@mail.xjtu.edu.cn; txy666793@stu.xjtu.edu.cn; nnzheng@mail.xjtu.edu.cn).

D. Cheng is with the Department of Computer Science, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: dcheng@xjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2852721

Many recent research works [4], [21]–[25] employ deep convolutional neural networks (DCNNs) trained by new loss functions, such as contrastive loss [26], [27], triplet loss [23], and so on, to learn features that can minimize the within-class variance and, meanwhile, maximize the between-class distance. However, both the contrastive and the triplet losses suffer from dramatic data expansion when constituting the sample pairs or sample triplets from the given training set. Moreover, it has been reported that the way of constituting pairs or triplets of training samples can significantly affect the performance accuracy of a DCNN model by a few percentage points [23], [28]. As a result, using such losses may lead to a slower model convergence, higher computational cost, increased training complexity, and uncertainty.

There have been research works that proposed new loss functions or specialized CNN architectures to employ the label relationships among different fine-grained classes [4], [5], [28], [29] or to explore the correlations among different parts of the input images [11]. These methods have achieved the state-of-the-art classification accuracies on various benchmark data sets of fine-grained image classification.

In this paper, we develop a fine-grained image classifier using a general DCNN. We try to improve the fine-grained image classification accuracy of a DCNN model from the following two aspects. First, to better model the h-level hierarchical label structure of the fine-grained image classes, we replace the top fully connected (fc) layer of a given DCNN model with the h fc layers that, each, correspond to the corresponding hierarchy of the layered label structure. Each of the h fc layers is fc to both its underneath layer and the feature output layer and is trained using a softmax loss with the labels from the corresponding label hierarchy. The h softmax losses used to train the h fc layers are called cascaded softmax loss, in this paper. Second, we propose a novel loss function, namely, the generalized large-margin (GLM) loss, which explicitly explores the hierarchical label structure and the similarity regularities of the fine-grained image classes. More specifically, for each given fine-grained class c, we divide the remaining fine-grained classes into two groups $\mathrm{SP}(c)$ and $\neg\mathrm{SP}(c)$, which consist of the fine-grained classes that share and do not share the same coarse (parent) class with c, respectively. The proposed GLM loss explicitly enforces that: 1) the distance between c and the nearest fine-grained class in $\mathrm{SP}(c)$ is larger than the within-class variance of c by a predefined margin and 2) the distance between c and its nearest fine-grained class in $\neg\mathrm{SP}(c)$ is larger than the distance between c and its farthest fine-grained class in $\mathrm{SP}(c)$ by a predefined margin. The first part of the GLM loss aims to make the DCNN models learn features that can reduce the within-class variance and maximize the between-class distance at the same time, while the second part is based on the common sense that the subclasses belonging to the same coarse class should be more similar to each other than those belonging to different coarse classes. Since GLM loss is differentiable, it can be used to train DCNNs with the standard backpropagation (BP) algorithm. Moreover, the proposed fine-grained image classification framework is independent and can be applied to any DCNN structures.

The main contributions of this paper are as follows.

1) We introduce the h fc layers to replace the top fc layer of a given DCNN model and train them with the cascaded softmax loss to better model the h-level hierarchical label structure of the fine-grained image classes.
2) We propose the GLM loss to make the given DCNN model explicitly explore the hierarchical label structure and the similarity regularities of the fine-grained image classes.
3) Comprehensive experimental evaluations of several general DCNN models using three benchmark data sets for the fine-grained image classification task demonstrate the effectiveness of the proposed framework.

The remaining of this paper is organized as follows. Section II reviews related works. Section III describes the methodology, including hierarchical label structure, DCNN modification and cascaded softmax loss, GLM loss, and the optimization of our framework. Section IV presents the experimental evaluations, and Section V concludes this paper.

II. Related Work

Methods to improve the fine-grained image classification accuracies can be broadly divided into the following three categories: 1) methods based on hand-crafted features; 2) part-based methods; and 3) metric learning-based methods. This section reviews representative works for each category.

A. Methods Based on Hand-Crafted Features

Krause *et al.* [1] proposed to use the spatial pyramid matching [30] in combination with the locality-constrained linear coding [31] to obtain the feature representations for fine-grained image classification. Lin *et al.* [11] implemented a fisher vector (FV)-scale-invariant feature transform (SIFT) method which first extracted the SIFT features [32] from each input image, then learned a Gaussian mixture model with these SIFT features to obtain the FVs of the input images, and finally trained a set of one-versus-all linear SVMs to classify the fine-grained image classes.

B. Part-Based Methods

A key challenge for fine-grained image classification is to recognize the subtle appearance differences between the images of similar fine-grained classes. Many part-based methods have been proposed to capture the subtle differences by localizing and representing discriminative object parts. Based on the deformable parts model, Zhang *et al.* [33] proposed the deformable parts descriptor, a pose-normalized descriptor, to facilitate the fine-grained image classification task. Chai *et al.* [34] proposed a symbiotic segmentation and parts localization model to classify the images with subtle differences. Krause *et al.* [35] proposed to detect important object parts and represent their appearances using the "ensemble of localized learned features." Branson *et al.* [17] developed a pose-normalized DCNN for fine-grained image classification. Zhang *et al.* [18] proposed a parts recognition model (named PR-CNN) by employing the R-CNN framework [36]. The main drawback of the above-mentioned methods is that they

need parts' annotations in the training process, which are significantly more expensive to collect than image labels. To explore the object parts information in an unsupervised way, Lin *et al.* [11] proposed a bilinear architecture (named Bilinear-CNN) that uses two separate DCNN feature extractors whose outputs are multiplied using outer product at each location of the produced feature maps and are pooled to obtain an image descriptor. To date, Bilinear-CNN has achieved the state-of-the-art classification accuracies on several benchmark data sets for fine-grained image classification. However, the employment of two parallel DCNNs has significantly increased the memory assumption and the training and testing costs of this method.

C. Metric Learning-Based Methods

Methods in this category attempt to learn a feature metric, such that the images from the same class are pulled closer, while those from different classes are pushed apart from each other in the learned feature space. Many loss functions have been proposed to improve CNNs' metric learning performances. To list some examples, the contrastive loss [26], [27] mandates that the distance between two image samples from different classes is larger than that of two samples from the same class by a predefined margin. The triplet loss [23] constitutes a large amount of triplets from the training set in the course of training. Each triplet contains an anchor sample A, a positive sample P, and a negative sample N, where A and P are from the same class, whereas A and N come from two different classes. It enforces that the distance between A and N must be larger than that between A and P by a predefined margin. When training CNN with the contrastive loss or triplet loss, they face the problems of dramatic data expansion, slow convergence, and instability, as described in Section I.

To address these problems, Wen *et al.* [28] proposed the center loss that simultaneously learns a center for each class and penalizes the distances between the learned feature vectors and their corresponding class centers. However, the center loss only considers the within-class compactness but does not consider the separability among different classes. This may cause such problems that different class centers become close to each other and may hamper DCNN models from learning truly discriminative features. Shi *et al.* [29], [37] proposed the min–max loss that explicitly enforces that the feature vectors learned by a DCNN model have the minimum with-class distances and the maximum between-class distances.

Zhou and Lin [5] proposed to exploit the label relationships among different fine-grained classes through bipartite-graph labels (BGLs). However, BGL can only handle the two-level hierarchical label structure and has no ability to be generalized to multilevel ones. Zhang *et al.* [4] proposed generalized triplet loss (GTL) for the hierarchical label structure. However, the GTL loss still has the above-mentioned problems of triplet loss.

Essentially, the contrastive, triplet, and GTL losses are all the point-to-point metric learning loss without considering the overall distribution of a training set. In contrast, the proposed GLM loss can be viewed as a set-to-set metric learning loss.

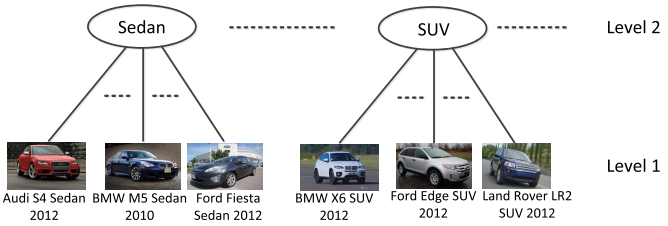


Fig. 2. Two-level hierarchical label structure in the Stanford car data set [1]. The fine-grained class labels in the leaf nodes are grouped into the coarse class labels according to their body types.

III. Methodology

A. Hierarchical Label Structure

In a typical benchmark data set for fine-grained image classification, class labels are grouped into a tree structure based on their semantics. Fig. 2 depicts the two-level label structure of the Stanford car data set [1], where the leaf and the root nodes correspond to the fine-grained and the coarse class labels, respectively. The fine-grained class labels represent the specific models from certain car makers (e.g., Audi S4 Sedan 2012, BMW M5 Sedan 2010, BMW X6 SUV 2012, and so on) which are grouped into coarse class labels according to their body types (e.g., Sedan, SUV, and so on).

An image data set with a hierarchical label structure can be mathematically defined as follows. Denote by $\mathcal{T} = \{\mathbf{X}_i, \mathbb{C}_i\}_{i=1}^{n}$ the set of training samples, where $\mathbf{X}_i$ denotes the ith sample image and n is the total number of training samples. Each sample image $\mathbf{X}_i$ is associated with a hierarchy of class labels $\mathbb{C}_i = \{c_i^j\}_{j=1}^{h}$, where $c_i^j \in \{1, 2, \ldots, C^{(j)}\}$ is its jth level class label, $C^{(j)}$ is the number of classes in level j, and h is the number of levels in the hierarchical label set. Assume that the fine-grained class labels are the first level class labels, i.e., c_i^1 is the fine-grained class label of the sample $\mathbf{X}_i$ and $C^{(1)}$ is the number of fine-grained classes. For image $\mathbf{X}_i$, we denote the output[1] of the penultimate layer of a DCNN by $\mathbf{x}_i$ and view $\mathbf{x}_i$ as the feature vector of $\mathbf{X}_i$ extracted by the network.

The proposed fine-grained image classification framework consists of the following two main components: 1) modify the network structure of a given DCNN and train it with the cascaded softmax loss and 2) develop the GLM loss. Sections II-B–II-D describe these two components in detail.

B. DCNN Modification and Cascaded Softmax Loss

For the fine-grained image classification problem with h-levels of class labels, we modify the given DCNN model by replacing its top fc layer with the h fc layers and train it using the cascaded softmax loss function. To ease the explanation, and without loss of generality, we describe the use of AlexNet [38] to classify an image data set with two levels of class labels. DCNN modifications for other fine-grained image classification problems can be derived by analogy.

The original AlexNet consists of five convolutional (conv) layers (conv1–5) and three fc layers (fc6–8) with fc7 and fc8 as

[1] Assume that the output has been reshaped into a column vector.

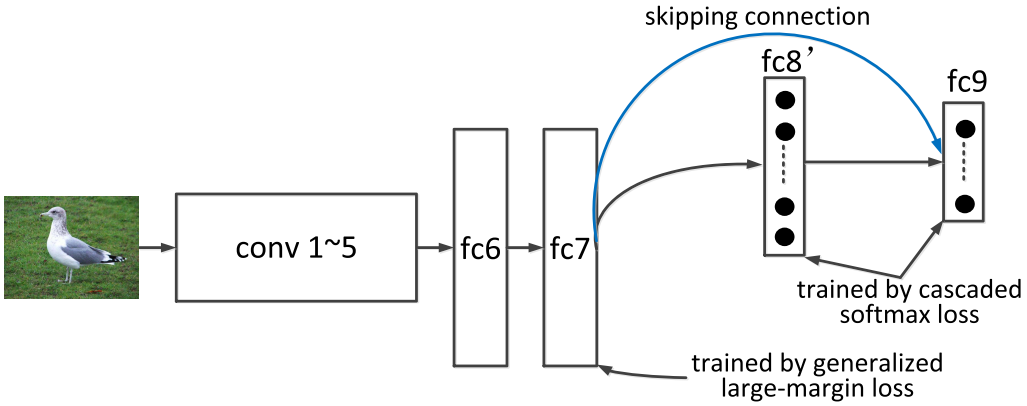


Fig. 3. Illustration of the proposed fine-grained image classification framework for the two-level label structure. The network shown in this figure is the modified AlexNet with the top fc8 layer replaced by fc8' and fc9. The output dimensions of fc8' and fc9 are equal to the number of fine-grained classes and the number of coarse classes in the data set, respectively. The blue connection refers to the skipping connection.

the feature output layer and the top fc layer, respectively. For classifying images with two levels of class labels, we replace fc8 with fc8' and fc9 and fully connect fc9 with both fc7 and fc8'. We call the connection from fc7 to fc9 as *skipping connection* (see Fig. 3). The numbers of neurons in fc8' and fc9 are set to $C^{(1)}$ and $C^{(2)}$, respectively. Given an input image $\mathbf{X}_i$, fc8' and fc9 output the probability scores $p(c_i^1|\mathbf{x}_i)$ of all the leaf fine-grained class labels $c_i^1 \in \{1, 2, \ldots, C^{(1)}\}$ and scores $p(c_i^2|\mathbf{x}_i)$ of all the coarse class labels $c_i^2 \in \{1, 2, \ldots, C^{(2)}\}$ for $\mathbf{X}_i$, respectively.

We introduce the skipping connection (fc7→fc9) to provide the coarse level classification layer (fc9) with access to both the learned features (output of fc7) and the predicted probability scores $p(c_i^1|\mathbf{x}_i)$ of all the fine-grained classes $c_i^1 \in \{1, 2, \ldots, C^{(1)}\}$ (output of fc8') for the input image $\mathbf{X}_i$. Intuitively, conducting the coarse level classification using the above-mentioned two types of information will be superior to the one using only the fine-grained level classification results, because the former explores both the semantics (i.e., the learned features) and the hierarchical label structure of the training samples. On the other hand, during the iterative training process, the prediction errors of fc9 are back propagated into fc8', fc7, and the lower layers of the network, which serves to gradually improve the prediction accuracy of fc8' as well.

Given the above-described modified AlexNet, we apply the cascaded softmax loss to fc8' and fc9 during the training process, which is defined as

$$\text{csm}(\mathcal{W}, \mathbf{X}_i, \mathbb{C}_i) = \sum_{j=1}^{h} \text{softmax}\left(\mathcal{W}, \mathbf{X}_i, c_i^j\right) \tag{1}$$

where $\mathcal{W}$ refers to the entire weight parameter set of the network. For classifying images with two levels of class labels, $h = 2$, $\text{softmax}(\mathcal{W}, \mathbf{X}_i, c_i^1)$ and $\text{softmax}(\mathcal{W}, \mathbf{X}_i, c_i^2)$ are applied to fc8' and fc9, respectively. Indeed, applying cascaded softmax loss could be regarded as a kind of multitask learning, where one task is for the fine-grained level classification and the other one for the coarse level classification. By sharing the feature representations between these two tasks, they can gradually improve each other in the process of joint training [39].

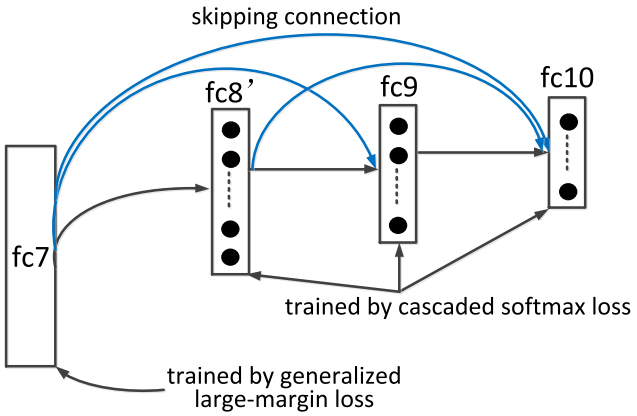


Fig. 4. Illustration of the proposed fine-grained image classification framework for the three-level label structure. The network shown in this figure is the modified AlexNet with the top fc8 layer replaced by fc8', fc9, and fc10. The output dimensions of fc8', f9, and f10 are equal to the number of fine-grained classes in the bottom layer and the numbers of coarse classes the level 2 and 3 layers, respectively. For simplicity, we omit the layers before fc7. The blue connections refer to the skipping connections.

The overall objective function to train the modified DCNN model is defined as

$$L = \frac{1}{n} \sum_{i=1}^{n} \text{csm}(\mathcal{W}, \mathbf{X}_i, \mathbb{C}_i) + \lambda \mathcal{M}(\mathcal{W}, \mathcal{X}, \mathcal{C}) \tag{2}$$

where $\text{csm}(\mathcal{W}, \mathbf{X}_i, \mathbb{C}_i)$ is the cascaded softmax loss defined in (1) and $\mathcal{M}(\mathcal{W}, \mathcal{X}, \mathcal{C})$ denotes the GLM loss to be applied to the feature output layer of the network (fc7 for AlexNet). The input to it includes $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ that is the set of extracted feature vectors and $\mathcal{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_n\}$ that is the set of hierarchical class labels for all the training samples. λ is the hyperparameter that controls the tradeoff between the cascaded softmax loss and the GLM loss. The derivation of the GLM loss and its gradient computation are provided in Sections III-C and III-D.

It is worth noting that Fig. 3 is only an illustration of the proposed framework that employs AlexNet for the two-level hierarchical label structure. This framework can be generalized to multilevel hierarchical label structure and is independent of any DCNN structure. Fig. 4 illustrates the modification of AlexNet for the three-level label structure. For simplicity,

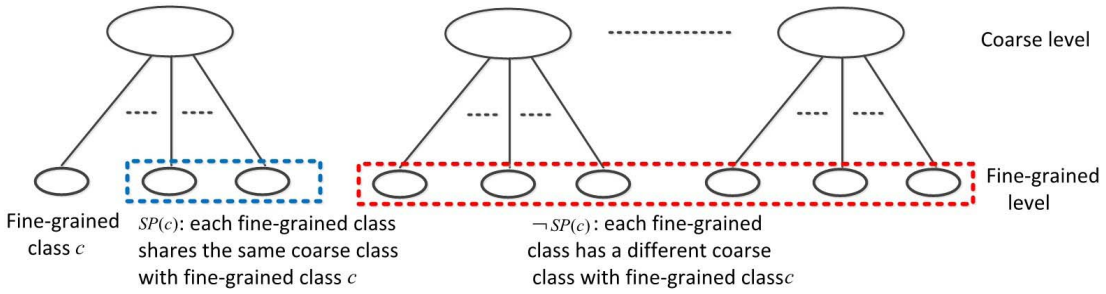


Fig. 5. Two-level hierarchical label structure. For a given fine-grained class c, all the remaining fine-grained classes are divided into two groups SP(c) (the classes in the blue dashed box) and ¬SP(c) (the classes in the red dashed box), which consist of the fine-grained classes that share and do not share the same coarse class with c, respectively.

we omit the layers before fc7 in Fig. 4. Compared to the original AlexNet, we replace fc8 with fc8', fc9, and fc10. fc9 is fc to fc7 and fc8', whereas fc10 is fc to fc7, fc8', and fc9. The output dimensions of fc8', fc9, and fc10 are equal to the number of fine-grained classes in the bottom layer and the numbers of the coarse classes in the layers 2 and 3, respectively.

C. Generalized Large-Margin Loss

For simplicity, we first derive the GLM loss for the two-level label structure and, then, generalize it to multiple levels.

For each given fine-grained class c, we divide the remaining fine-grained classes into two groups SP(c) and ¬SP(c) which consist of the fine-grained classes that share and do not share the same parent coarse class with c, respectively (see Fig. 5). The proposed GLM loss explicitly enforces that: 1) the distance between c and the nearest fine-grained class in SP(c) is larger than the within-class distance of c by a predefined margin and 2) the distance between c and its nearest fine-grained class in ¬SP(c) is larger than the distance between c and its farthest fine-grained class in SP(c) by a predefined margin. In the following part of this section, we will first define the within-class variance and between-class distance and, then, derive the GLM loss using these definitions.

Denote the feature vector set of the training samples belonging to the fine-grained class c by

$$\mathcal{S}_c = \{\mathbf{x}_i | i \in \pi_c\} \tag{3}$$

where π_c is the index set of the training samples belonging to class c. The mean vector of $\mathcal{S}_c$ can be represented as

$$\mathbf{m}_c = \frac{1}{n_c} \sum_{i \in \pi_c} \mathbf{x}_i \tag{4}$$

where $n_c = |\pi_c|$. The within-class distance function [40] for $\mathcal{S}_c$ can be represented as

$$D^{(W)}(\mathcal{S}_c) = \frac{1}{n_c} \sum_{i \in \pi_c} \|\mathbf{x}_i - \mathbf{m}_c\|^2. \tag{5}$$

Let $\mathcal{S}_p$ and $\mathcal{S}_q$ be the two feature vector sets defined as

$$\mathcal{S}_p = \{\mathbf{x}_i | i \in \pi_p\}, \quad \mathcal{S}_q = \{\mathbf{x}_i | i \in \pi_q\}. \tag{6}$$

The between-class distance of $\mathcal{S}_p$ and $\mathcal{S}_q$ can be expressed as

$$\begin{aligned} D^{(B)}(\mathcal{S}_p, \mathcal{S}_q) &= \frac{1}{2k} \sum_{i,j=1}^{n} G_{ij}^{(p,q)} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \\ &= \frac{1}{k} tr(\mathbf{H} \boldsymbol{\Psi}^{(p,q)} \mathbf{H}^\top) \end{aligned} \tag{7}$$

$$G_{ij}^{(p,q)} = \begin{cases} 1, & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \zeta_k(\mathcal{S}_p, \mathcal{S}_q) \text{ or } \zeta_k(\mathcal{S}_q, \mathcal{S}_p) \\ 0, & \text{else} \end{cases} \tag{8}$$

where $G_{ij}^{(p,q)}$ denotes the element (i, j) of the affinity matrix $\mathbf{G}^{(p,q)}$ between $\mathcal{S}_p$ and $\mathcal{S}_q$, $\mathbf{G}^{(p,q)} = (G_{ij}^{(p,q)})_{n \times n}$, $\zeta_k(\mathcal{S}_p, \mathcal{S}_q)$ is the set that consists of the k-nearest sample pairs from the sample pair set $\{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \in \mathcal{S}_p, \mathbf{x}_j \in \mathcal{S}_q\}$, $\mathbf{H} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, $\boldsymbol{\Psi}^{(p,q)}$ is the Laplacian matrix of $\mathbf{G}^{(p,q)}$, i.e., $\boldsymbol{\Psi}^{(p,q)} = \mathbf{D}^{(p,q)} - \mathbf{G}^{(p,q)}$, $\mathbf{D}^{(p,q)} = \text{diag}(d_{11}^{(p,q)}, \ldots, d_{nn}^{(p,q)})$, $d_{ii}^{(p,q)} = \sum_{j=1, j \neq i}^{n} G_{ij}^{(p,q)}$, $i = 1, 2, \ldots, n$, and tr($\cdot$) denotes the trace of a matrix.

Given the above-mentioned definitions, the two constraints of the GLM loss can be derived as

$$\mathcal{M}_c^{(1)} = \left[D^{(W)}(\mathcal{S}_c) - D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{\text{SP}(c)}^{(\min)}\right) + \alpha_1\right]_+ \tag{9}$$

$$\mathcal{M}_c^{(2)} = \left[D^{(B)}(\mathcal{S}_c, \mathcal{S}_{\text{SP}(c)}^{(\max)}) - D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{\neg\text{SP}(c)}^{(\min)} + \alpha_2\right]_+ \tag{10}$$

where α_1 and α_2 are the two predefined margins, $[x]_+ = \max\{x, 0\}$, and

$$\mathcal{S}_{\text{SP}(c)}^{(\min)} = \arg \min_{\mathcal{S}_i, i \in \text{SP}(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i) \tag{11}$$

$$\mathcal{S}_{\text{SP}(c)}^{(\max)} = \arg \max_{\mathcal{S}_i, i \in \text{SP}(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i) \tag{12}$$

$$\mathcal{S}_{\neg\text{SP}(c)}^{(\min)} = \arg \min_{\mathcal{S}_i, i \in \neg\text{SP}(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i). \tag{13}$$

In (11)–(13), SP(c) consists of the fine-grained classes that share the same parent coarse class with c, $\mathcal{S}_{\text{SP}(c)}^{(\min)}$ is the feature vector set of the training samples belonging to the fine-grained class in SP(c) that is the closest to c, and $\mathcal{S}_{\text{SP}(c)}^{(\max)}$ is the feature vector set of the training samples belonging to the fine-grained class in SP(c) that is the farthest to c. Furthermore, ¬SP(c) consists of the fine-grained classes that do not share the same parent coarse class with c and $\mathcal{S}_{\neg\text{SP}(c)}^{(\min)}$ is the feature vector set of the training samples belonging to the fine-grained class in ¬SP(c) that is the closest to c (see Fig. 5).

Using the above-mentioned definitions, the GLM loss for the two-level label structure can be defined as

$$\mathcal{M} = \frac{1}{C^{(1)}} \sum_{c=1}^{C^{(1)}} \left(\mathcal{M}_c^{(1)} + \mathcal{M}_c^{(2)}\right). \tag{14}$$

So far, we have mainly discussed the scenario of the two-level label structure. In fact, the GLM loss can be easily extended to more general multilevel cases. For instance, the GLM loss for the three-level label structure can be derived as follows. First, for each given fine-grained class c, in addition to dividing the remaining fine-grained classes into the two groups $\mathrm{SP}(c)$ and $\neg\mathrm{SP}(c)$, we further divide them into $SG(c)$ and $\neg SG(c)$, which consist of the fine-grained classes that share and do not share the same grandparent coarse class with c, respectively. Next, we add the third constraint into the GLM loss: *the distance between c and its nearest fine-grained class in* $\neg SG(c)$ *is larger than the distance between c and its farthest fine-grained class in* $SG(c)$ *by a predefined margin*. This statement can be mathematically expressed as

$$\mathcal{M}_c^{(3)} = \left[D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{SG(c)}^{(\max)}\right) - D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{\neg SG(c)}^{(\min)}\right) + \alpha_3\right]_+ \tag{15}$$

where α_3 is the predefined margin, and

$$\mathcal{S}_{SG(c)}^{(\max)} = \arg \max_{\mathcal{S}_i, \in SG(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i) \tag{16}$$

$$\mathcal{S}_{\neg SG(c)}^{(\min)} = \arg \min_{\mathcal{S}_i, i \in \neg SG(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i). \tag{17}$$

Using the above-mentioned definitions, the GLM loss for the three-level label structure can be defined as

$$\mathcal{M} = \frac{1}{C^{(1)}} \sum_{c=1}^{C^{(1)}} \left(\mathcal{M}_c^{(1)} + \mathcal{M}_c^{(2)} + \mathcal{M}_c^{(3)}\right) \tag{18}$$

where $\mathcal{M}_c^{(1)}$ and $\mathcal{M}_c^{(2)}$ are defined by (9) and (10), respectively.

D. Optimization

We use the standard BP algorithm with mini-batch to train the modified DCNN model. The overall objective function is (2). Hence, we need to compute the gradients of L with respect to the activations of all the DCNN layers which are called the error flows of the corresponding layers. The gradient calculation of softmax loss is straightforward. In the following, we only provide the gradient of the GLM loss with respect to $\mathbf{x}_i$.

For the two-level hierarchical label structure, the derivatives of $\mathcal{M}$ with respect to $\mathbf{x}_i$ can be computed by

$$\frac{\partial \mathcal{M}}{\partial \mathbf{x}_i} = \frac{1}{C^{(1)}} \sum_{c=1}^{C^{(1)}} \left(\frac{\partial \mathcal{M}_c^{(1)}}{\partial \mathbf{x}_i} + \frac{\partial \mathcal{M}_c^{(2)}}{\partial \mathbf{x}_i}\right) \tag{19}$$

$$\frac{\partial \mathcal{M}_c^{(1)}}{\partial \mathbf{x}_i} = \delta_1(c) \left[I(i \in \pi_c) \frac{2}{n_c} \left(1 - \frac{1}{n_c}\right)(\mathbf{x}_i - \mathbf{m}_c) - \frac{2}{k} \left(\mathbf{H} \boldsymbol{\Psi}^{\left(c, c_{\mathrm{SP}}^{(\min)}\right)}\right)_{(:,i)} \right] \tag{20}$$

$$\frac{\partial \mathcal{M}_c^{(2)}}{\partial \mathbf{x}_i} = \delta_2(c) \frac{2}{k} \mathbf{H} \left[\boldsymbol{\Psi}^{\left(c, c_{\mathrm{SP}}^{(\max)}\right)} - \boldsymbol{\Psi}^{\left(c, c_{\neg \mathrm{SP}}^{(\min)}\right)}\right]_{(:,i)} \tag{21}$$

Algorithm 1 Training Algorithm for Our Framework Shown in Fig. 3

Input: Training set $\mathcal{T}$, hyperparameters λ, α_1 and α_2, maximum number of iterations I_{max}, and counter $iter = 0$.
Output: $\mathcal{W}$.

1: Select a training mini-batch from $\mathcal{T}$.
2: Perform the forward propagation, for each sample, computing the activations of all layers.
3: Compute the error flows of fc9 from the softmax loss (for coarse classes). Then compute the error flows of layer fc7 and fc8' from layer fc9 by backward propagation, respectively.
4: Compute the error flows of fc8' from the softmax loss (for fine-grained classes).
5: Compute the total error flows of fc8', which is the summation of those from fc9 and the softmax loss (for fine-grained classes). Then compute the error flows of layer fc7 from layer fc8' by the BP algorithm.
6: Calculate the error flows of layer fc7 from the GLM loss according to Eq. (19) and the scaling factor λ.
7: Calculate the total error flows of layer fc7, which is the summation of those from fc8', fc9 and GLM loss.
8: Perform the back-propagation from layer fc7 to layer conv1, sequentially computing the error flows of these layers by BP algorithm.
9: According to the activations and error flows of all layers, compute $\frac{\partial \mathcal{L}}{\partial \mathcal{W}}$ by BP algorithm.
10: Update $\mathcal{W}$ by gradient descent algorithm.
11: $iter \leftarrow iter + 1$. If $iter < I_{max}$, perform step 1.

where $I(\cdot)$ denotes the indicator function that equals one if the condition is true, and zero otherwise. The subscript symbol $(:, i)$ denotes the ith column of a matrix, and

$$\delta_1(c) = \begin{cases} 1, & D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{\mathrm{SP}(c)}^{(\min)}\right) - D^{(W)}(\mathcal{S}_c) < \alpha_1 \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

$$\delta_2(c) = \begin{cases} 1, & D\left(\mathcal{S}_c, \mathcal{S}_{\mathrm{SP}(c)}^{(\max)}, \mathcal{S}_{\neg \mathrm{SP}(c)}^{(\min)}\right) < \alpha_2, \\ 0, & \text{otherwise}, \end{cases} \tag{23}$$

$$D\left(\mathcal{S}_c, \mathcal{S}_{\mathrm{SP}(c)}^{(\max)}, \mathcal{S}_{\neg \mathrm{SP}c}^{(\min)}\right) = D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{\neg \mathrm{SP}(c)}^{(\min)}\right) - D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{\mathrm{SP}(c)}^{(\max)}\right) \tag{24}$$

$$c_{\mathrm{SP}}^{(\min)} = \arg \min_{i \in \mathrm{SP}(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i) \tag{25}$$

$$c_{\mathrm{SP}}^{(\max)} = \arg \max_{i \in \mathrm{SP}(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i) \tag{26}$$

$$c_{\neg \mathrm{SP}}^{(\min)} = \arg \min_{i \in \neg \mathrm{SP}(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i). \tag{27}$$

For the three-level hierarchical label structure, the derivatives of $\mathcal{M}$ with respect to $\mathbf{x}_i$ can be computed by

$$\frac{\partial \mathcal{M}}{\partial \mathbf{x}_i} = \frac{1}{C^{(1)}} \sum_{c=1}^{C^{(1)}} \left(\frac{\partial \mathcal{M}_c^{(1)}}{\partial \mathbf{x}_i} + \frac{\partial \mathcal{M}_c^{(2)}}{\partial \mathbf{x}_i} + \frac{\partial \mathcal{M}_c^{(3)}}{\partial \mathbf{x}_i}\right) \tag{28}$$

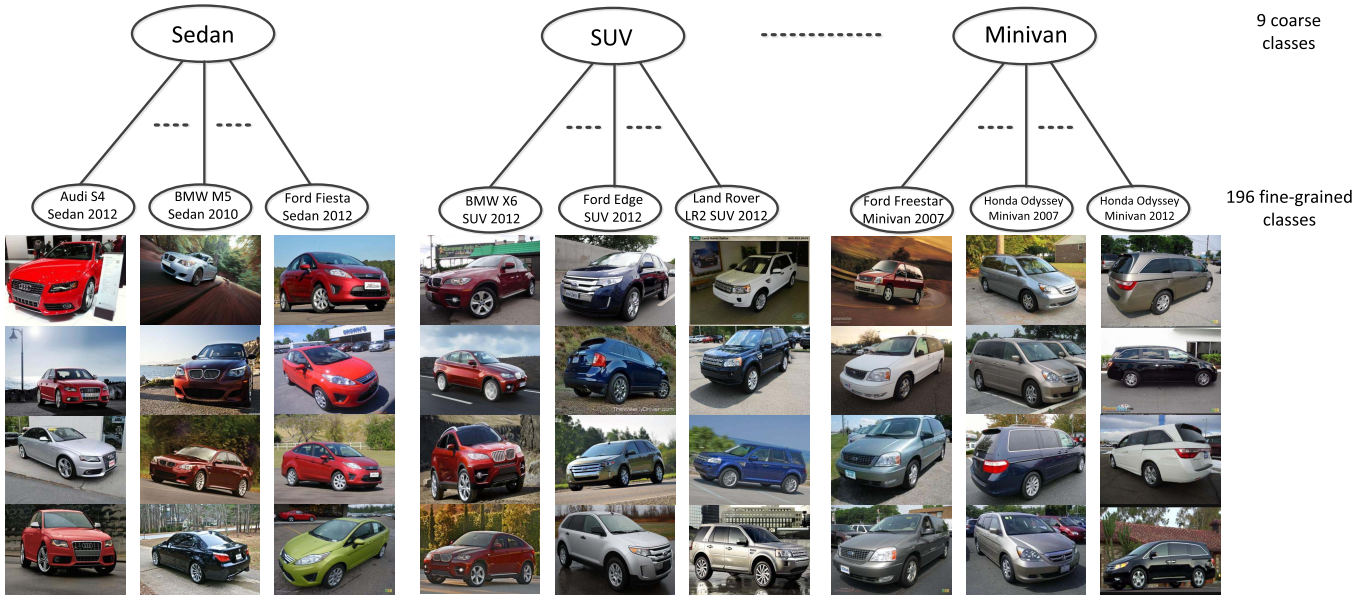


Fig. 6. Sample images from the Stanford car data set with the two-level hierarchical label structure.

where $(\partial \mathcal{M}_c^{(1)}/\partial \mathbf{x}_i)$ and $(\partial \mathcal{M}_c^{(2)}/\partial \mathbf{x}_i)$ are defined by (20) and (21), respectively, and

$$\frac{\partial \mathcal{M}_c^{(3)}}{\partial \mathbf{x}_i} = \delta_3(c)\frac{2}{k}\mathbf{H}\left[\mathbf{\Psi}^{\left(c,c_{SG}^{(\max)}\right)} - \mathbf{\Psi}^{\left(c,c_{\neg SG}^{(\min)}\right)}\right]_{(:,i)} \tag{29}$$

$$\delta_3(c) = \begin{cases} 1, & D\left(\mathcal{S}_c, \mathcal{S}_{SG(c)}^{(\max)}, \mathcal{S}_{\neg SG(c)}^{(\min)}\right) < \alpha_3 \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

$$D\left(\mathcal{S}_c, \mathcal{S}_{SG(c)}^{(\max)}, \mathcal{S}_{\neg SG(c)}^{(\min)}\right) = D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{\neg SG(c)}^{(\min)}\right) - D^{(B)}\left(\mathcal{S}_c, \mathcal{S}_{SG(c)}^{(\max)}\right) \tag{31}$$

$$c_{SG}^{(\max)} = \arg\max_{i \in SG(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i) \tag{32}$$

$$c_{\neg SG}^{(\min)} = \arg\min_{i \in \neg SG(c)} D^{(B)}(\mathcal{S}_c, \mathcal{S}_i). \tag{33}$$

In Algorithm 1, based on (19), we provide the training algorithm for our framework shown in Fig. 3. It is worth noting that Algorithm 1 is only an example of the proposed framework for two-level label structure based on the eight-layer AlexNet model. Based on (28), one can easily give corresponding training algorithm for three-level label structure.

IV. Experimental Evaluations

A. Overall Settings

To reveal the effectiveness and generality of the proposed fine-grained image classification method, we conduct comprehensive experimental evaluations on three image data sets with hierarchical label structures, i.e., Stanford car [1], fine-grained visual classification (FGVC)-aircraft [10], and CUB-200-2011 [6] using three popular DCNNs of different network complexities, i.e., AlexNet [38], GoogLeNet [41], and VGG [42]. AlexNet contains 5 conv and 3 fc layers, GoogLeNet contains 22 conv and 1 fc layers, while VGG contains 16 conv and 3 fc layers. All the three DCNN models are pretrained using the ImageNet data set [43].[2] All the experiments are conducted using the Caffe platform [44]. For those hyperparameters, including drop ratio, momentum, and weight decay, we strictly follow the original network settings.

The proposed method consists of two novel components: 1) DCNN modification with the skipping connections trained by the cascaded softmax loss and 2) the GLM loss. To reveal how each component contributes to the performance improvement, we implement the following six variants for each given network and data set.

1) *XXX-SM:* The original XXX network is trained using the standard softmax loss with the fine-grained class labels.
2) *XXX-SM-GLM:* The original XXX network is trained using the standard softmax loss with the fine-grained class labels and the proposed GLM loss.
3) *XXX-CSM:* The modified XXX network without the skipping connections is trained using the cascaded softmax loss.
4) *XXX-CSM-GLM:* The modified XXX network without the skipping connections is trained using the cascaded softmax loss and the proposed GLM loss.
5) *XXX-SC-CSM:* The modified XXX network with the skipping connections is trained using the cascaded softmax loss.
6) *XXX-OURS:* The modified XXX network with the skipping connections is trained using both the cascaded softmax and the GLM losses.

For simplicity, we set all the margins $\alpha_1, \alpha_2, \alpha_3$ in the GLM loss to one. To obtain the optimal parameter value of λ, we tuned it on a validation set. Specifically, from the Stanford car data set [1], we first randomly select 1000 training images to form the validation set. Then, we use the remaining training

[2]All the DCNN-based methods used in our comparative studies pretrained their DCNN models using ImageNet. For fair comparisons, we follow the same convention by pretraining our model with ImageNet as well.

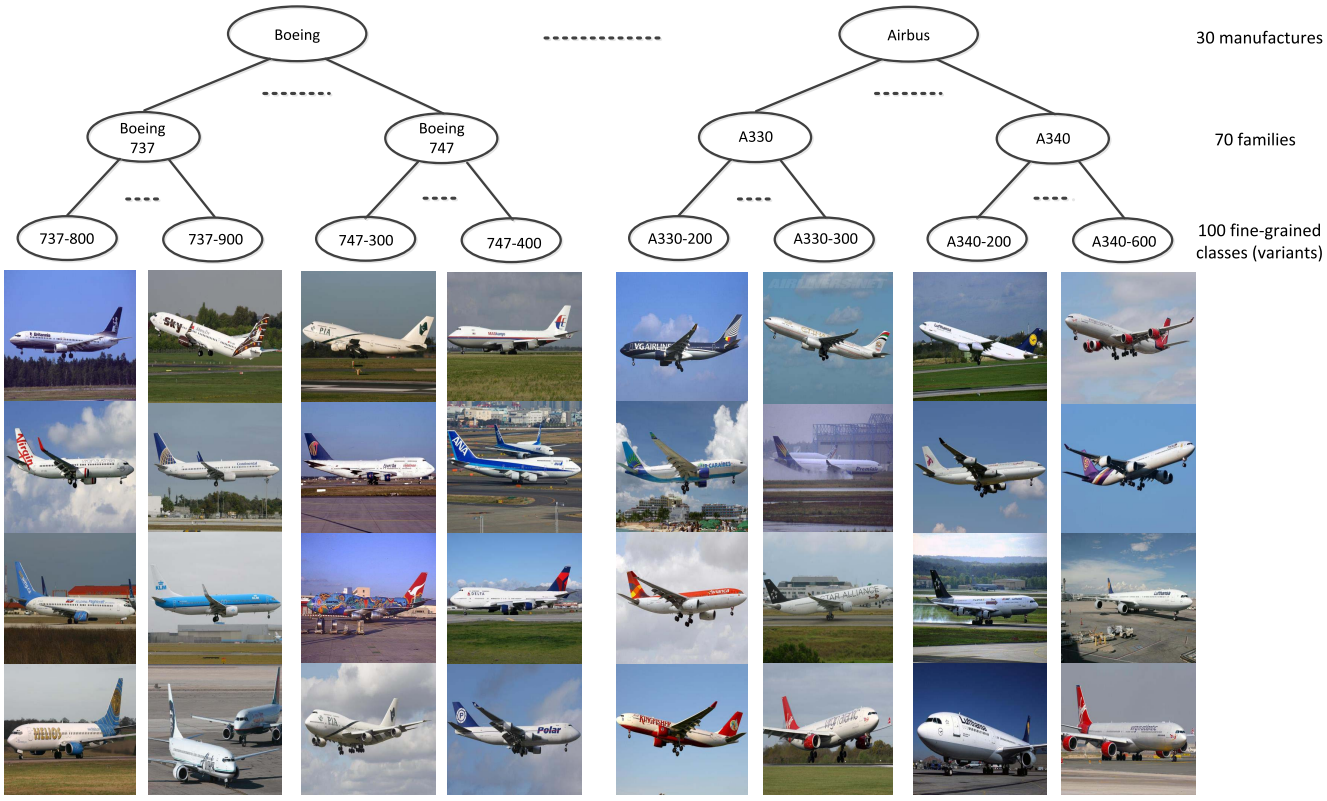


Fig. 7. Sample images from the FGVC-aircraft data set with the three-level hierarchical label structure.

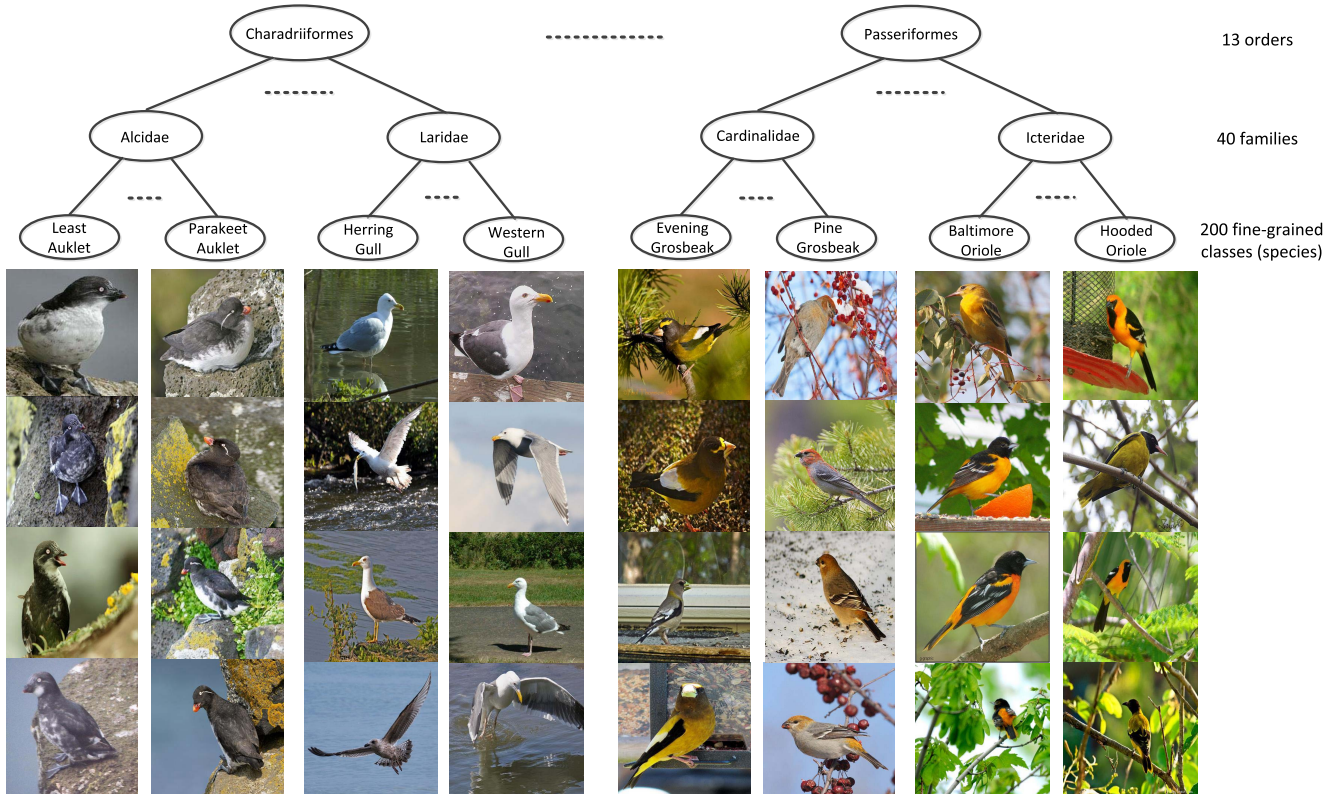


Fig. 8. Sample images from the CUB-200-2011 data set with the three-level hierarchical label structure.

images to train the modified AlexNet (i.e., AlexNet-SM-GLM) and use the validation set to tune the hyperparameter λ. After the hyperparameter λ is determined on the validation set ($\lambda = 0.8$), we fix it and use it for all the three DCNN models (AlexNet, GoogLeNet, and VGG) and the three image data sets (Stanford car, FGVC-aircraft, and CUB-200-2011 data

sets). All the three image data sets contain the ground-truth bounding box for each target object. For consistency with the previous studies, we also evaluated the proposed method with two experimental settings. In the first setting, we train and test the proposed method with each uncropped sample without using the ground-truth bounding box (BBox in short) annotations, whereas in the second setting, we train and test the proposed method using the image patches cropped from the ground-truth bounding boxes assigned to each sample.

B. Data Sets

1) Stanford Car Data Set [1]: It contains 16 185 images of 196 vehicle classes and is split into 8144 training and 8041 testing images, where each class is roughly in a 50–50 split. The data set consists of two layers of class labels, where the bottom layer contains 196 fine-grained classes representing specific models from certain car makers, such as Audi S4 Sedan 2012, BMW X6 SUV 2012, and so on, whereas the top layer contains nine coarse classes representing vehicle body types, such as Sedan, SUV, Coupe, and so on. Some sample images from this data set are shown in Fig. 6.

2) FGVC-Aircraft Data Set [10]: It contains 10 000 images of various aircrafts, which is organized into a three-layer label structure. The three layers, from bottom to top, consist of 100 aircraft variants (e.g., Boeing 737-700, Boeing 737-900), 70 families (e.g., Boeing 737, Boeing 747), and 30 manufacturers (e.g., Boeing and Airbus), respectively. Each aircraft variant (fine-grained image class) contains exactly 100 images. We adopt the standard training/test data split of 6667/3333 images. Sample images from the FGVC-aircraft data set are shown in Fig. 7.

3) CUB-200-2011 Data Set [6]: This data set contains 11 788 images of 200 bird species, which are grouped into the three-level label structure, with 200 species in the bottom layer, 40 families in the second layer, and 13 orders in the third layer. We adopt the standard training/test data split of 5994/5794 images. Sample images from this data set are shown in Fig. 8.

C. Experimental Evaluation Results

Tables I–III show the top-1 classification accuracies of all the methods under the comparison on the Stanford car, FGVC-aircraft, and CUB-200-2011 test sets, respectively. In Tables I–III, we have included the representative methods described in Section II, and used the same abbreviations of these methods as in Section II to report their experimental results.

Moreover, we also compare the proposed GLM loss with the contrastive loss [27], the triplet loss [23], the center loss [28], and the min–max loss [29], as mentioned in Section II. In Tables I–III, "XXX-SM-contrastive," "XXX-SM-triplet," "XXX-SM-center loss," and "XXX-SM-min–max" correspond to the XXX network trained with the softmax + contrastive loss, softmax + triplet loss, softmax + center loss, and softmax + min–max loss, respectively. It is worth noting that in the original papers of the contrastive loss, triplet loss, center loss, and min–max loss, there are no reports of the classification accuracies on these three image data sets. The results of these four methods are produced based on our own implementations.

TABLE I
TOP-1 CLASSIFICATION ACCURACIES (%) ON STANFORD CAR TEST SET

Method	**Without** BBox	**With** BBox
FV-SIFT [11]	51.2	−−
LLC [1]	−−	69.5
ELLF [35]	−−	73.9
AlexNet-BGL [5]	60.6	77.0
GoogLeNet-BGL [5]	79.4	87.1
VGG-BGL [5]	83.8	89.6
GoogLeNet-GTL [4]	−−	88.4
AlexNet-SM-contrastive [27]	59.4	76.1
AlexNet-SM-triplet [23]	59.8	76.3
AlexNet-SM-center loss [28]	61.1	76.8
AlexNet-SM-min-max [29]	61.5	77.2
GoogLeNet-SM-contrastive [27]	78.7	86.9
GoogLeNet-SM-triplet [23]	78.9	86.9
GoogLeNet-SM-center loss [28]	80.0	87.1
GoogLeNet-SM-min-max [29]	80.8	87.3
VGG-SM-contrastive [27]	86.3	89.7
VGG-SM-triplet [23]	86.5	89.8
VGG-SM-center loss [28]	86.9	89.9
VGG-SM-min-max [29]	86.9	90.2
Bilinear-CNN [D, D] [11]	90.6	−−
Bilinear-CNN [D, D]	90.4	90.7
Bilinear-CNN [D, D]-OURS	**91.8**	**92.0**
AlexNet-SM	59.0	76.0
AlexNet-SM-GLM	63.6	78.9
AlexNet-CSM	60.3	77.0
AlexNet-CSM-GLM	64.2	79.7
AlexNet-SC-CSM	61.6	77.8
AlexNet-OURS	**65.2**	**80.4**
GoogLeNet-SM	78.5	86.7
GoogLeNet-SM-GLM	83.0	89.7
GoogLeNet-CSM	79.9	87.9
GoogLeNet-CSM-GLM	84.0	90.1
GoogLeNet-SC-CSM	81.0	88.4
GoogLeNet-OURS	**85.0**	**90.5**
VGG-SM	85.9	89.5
VGG-SM-GLM	88.6	90.7
VGG-CSM	86.7	89.9
VGG-CSM-GLM	89.1	90.8
VGG-SC-CSM	87.2	90.2
VGG-OURS	**89.5**	**91.1**

From Tables I–III, we can see that "VGG-OURS" beats all the other methods except for Bilinear-CNN [D, D] [11], and that "Bilinear-CNN [D, D]-OURS" achieves the best classification accuracies on all the three benchmark test sets. Bilinear-CNN [D, D] uses two parallel VGG nets, whereas VGG-OURS only use a single VGG net. The employment of two parallel VGG nets by Bilinear-CNN [D, D] results in the significant increases of the memory assumption and the training and testing times, compared to VGG-OURS. It is noteworthy that our method improves the given DCNNs on learning the hierarchical class label structures by modifying the last fc layer and employing the proposed CSM + GLM loss functions, whereas Bilinear-CNN [D, D] [11] employs the two separate VGG feature extractors and the outer product of their output features to explore correlations among different parts of the input image. Therefore, these two methods can be combined to further improve performance accuracies. We have applied our proposed method to Bilinear-CNN [D, D] and

TABLE II
TOP-1 CLASSIFICATION ACCURACIES (%) ON FGVC-AIRCRAFT TEST SET

Method	**Without** BBox	**With** BBox
FV-SIFT [11]	55.7	−−
SSPL [34]	−−	72.5
AlexNet-SM-contrastive [27]	68.1	74.0
AlexNet-SM-triplet [23]	68.5	74.3
AlexNet-SM-center loss [28]	69.1	74.9
AlexNet-SM-min-max [29]	69.0	75.1
GoogLeNet-SM-contrastive [27]	77.0	81.4
GoogLeNet-SM-triplet [23]	77.2	81.4
GoogLeNet-SM-center loss [28]	77.7	81.9
GoogLeNet-SM-min-max [29]	78.1	82.2
VGG-SM-contrastive [27]	81.8	84.4
VGG-SM-triplet [23]	81.8	84.7
VGG-SM-center loss [28]	82.4	85.1
VGG-SM-min-max [29]	82.7	85.5
Bilinear-CNN [D, D] [11]	84.1	−−
Bilinear-CNN [D, D]	83.9	84.4
Bilinear-CNN [D, D]-OURS	**85.1**	**85.8**
AlexNet-SM	67.3	73.5
AlexNet-SM-GLM	70.5	75.8
AlexNet-CSM	68.3	74.6
AlexNet-CSM-GLM	71.2	76.6
AlexNet-SC-CSM	69.2	75.3
AlexNet-OURS	**72.0**	**77.1**
GoogLeNet-SM	76.4	81.1
GoogLeNet-SM-GLM	79.5	83.2
GoogLeNet-CSM	77.5	82.0
GoogLeNet-CSM-GLM	80.0	83.8
GoogLeNet-SC-CSM	78.0	82.8
GoogLeNet-OURS	**80.4**	**84.3**
VGG-SM	81.3	83.9
VGG-SM-GLM	84.0	86.7
VGG-CSM	82.2	84.8
VGG-CSM-GLM	84.2	87.0
VGG-SC-CSM	83.0	85.6
VGG-OURS	**84.6**	**87.5**

TABLE III
TOP-1 CLASSIFICATION ACCURACIES (%) ON CUB-200-2011 TEST SET

Method	**Without** BBox	**With** BBox
FV-SIFT [11]	12.8	24.1
DPD [33]	−−	51.0
PN-DCN [17]	−−	75.7
PR-CNN [18]	73.9	76.4
AlexNet-BGL [5]	58.2	65.5
GoogLeNet-BGL [5]	74.7	78.5
VGG-BGL [5]	74.2	79.4
AlexNet-SM-contrastive [27]	57.4	63.7
AlexNet-SM-triplet [23]	57.9	64.0
AlexNet-SM-center loss [28]	58.0	64.8
AlexNet-SM-min-max [29]	58.5	64.8
GoogLeNet-SM-contrastive [27]	74.1	77.8
GoogLeNet-SM-triplet [23]	74.1	78.0
GoogLeNet-SM-center loss [28]	74.5	78.4
GoogLeNet-SM-min-max [29]	75.1	78.9
VGG-SM-contrastive [27]	73.0	79.0
VGG-SM-triplet [23]	73.4	79.3
VGG-SM-center loss [28]	73.5	79.7
VGG-SM-min-max [29]	73.8	80.1
Bilinear-CNN [D, D] [11]	84.0	84.8
Bilinear-CNN [D, D]	84.1	84.8
Bilinear-CNN [D, D]-OURS	**85.4**	**85.7**
AlexNet-SM	56.9	63.0
AlexNet-SM-GLM	60.5	67.1
AlexNet-CSM	58.2	64.6
AlexNet-CSM-GLM	61.1	67.6
AlexNet-SC-CSM	59.3	65.5
AlexNet-OURS	**61.7**	**68.3**
GoogLeNet-SM	73.6	77.4
GoogLeNet-SM-GLM	76.8	80.5
GoogLeNet-CSM	74.6	78.4
GoogLeNet-CSM-GLM	77.1	81.3
GoogLeNet-SC-CSM	75.3	79.0
GoogLeNet-OURS	**77.6**	**82.0**
VGG-SM	72.5	78.6
VGG-SM-GLM	75.8	81.8
VGG-CSM	73.4	79.5
VGG-CSM-GLM	76.5	82.0
VGG-SC-CSM	73.9	80.0
VGG-OURS	**77.0**	**82.4**

included the experimental results in Tables I–III, respectively. In Tables I–III, "Bilinear-CNN [D, D] [11]" refers to the experimental results reported by the original paper [11], "Bilinear-CNN [D, D]" refers to our own experimental results, and "Bilinear-CNN [D, D]-OURS" refers to the experimental results of combining our method with Bilinear-CNN [D, D]. It can be seen that our method can further improve the state-of-the-art Bilinear-CNN [D, D] by up to 1.4 percentage points.

The evaluation results shown in Tables I–III can be summarized as follows.

1) Training a DCNN model with the GLM loss can get up to 4.6% performance improvement in comparison with the same model trained without using this loss, and the GLM loss is superior to the contrastive loss, triplet loss, center loss, and min–max loss.
2) Compared to an original DCNN model trained by the standard softmax loss, modifying the model with the skipping connections and training it with the cascaded softmax loss improve the classification accuracies by up to 2.6%.
3) From the experimental results of XXX-CSM and XXX-SC-CSM, it is clear that adding the skipping connections to the final fc layers of the given DCNN model can contribute to the performance improvement by up to another 1.3%.
4) Compared to GoogLeNet-SM on the Stanford car data set, GoogLeNet-OURS can improve the top-1 classification accuracy by up to 6.5%, which is significant.
5) The "Bilinear-CNN [D, D]-OURS" model achieves the highest classification accuracies on all the three benchmark test sets. Despite the very high state-of-the-art classification accuracies achieved by the Bilinear-CNN structure, our proposed method can still improve its classification accuracies by up to 1.4%.

The above-mentioned results illustrate the effectiveness of our method and show that our method can be applied to different DCNN models.

D. Sensitivity Study of λ

We conduct the sensitivity study of the hyperparameter λ to see whether the network performance changes a lot with a change of λ. To save the time and computational resources, we conduct this study only with AlexNet and the Stanford car data set. Specifically, we set λ to values chosen from the predefined range, train AlexNet with these parameter values on the Stanford car training set, and then report the top-1 classification accuracies on the Stanford car test set.

TABLE IV
INFLUENCE OF PARAMETER λ ON ALEXNET-SM-GLM WITH STANFORD CAR DATA SET. $\lambda = 0.0$ REFERS TO THE ALEXNET-SM BASELINE

λ	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6
Accuracy	59.0	62.4	62.8	63.1	63.6	63.8	62.6	62.1	61.0

We run AlexNet-SM-GLM on the Stanford car data set with λ varying from 0.2 to 1.4 with step size of 0.2. The top-1 classification accuracies are shown in Table IV. From Table IV, it can be observed that the performance does not change much by varying the value of λ.

V. CONCLUSION

A novel DCNN-based framework is proposed to improve fine-grained image classification accuracy. We improve the fine-grained image classification accuracy of a DCNN model from the following two aspects. First, we introduce h fc layers to replace the top fc layer of a given DCNN model and train them with the cascaded softmax loss to better model the h-level hierarchical label structure of the fine-grained image classes. Second, we propose the GLM loss to make the given DCNN model explicitly explore the hierarchical label structure and the similarity regularities of the fine-grained image classes. The proposed fine-grained image classification framework is independent of the DCNN structure. Comprehensive experimental evaluations of several general DCNN models using three benchmark data sets for the fine-grained image classification task demonstrate the effectiveness of our method.

REFERENCES

[1] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Jun. 2013, pp. 554–561.

[2] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis, "Jointly optimizing 3D model fitting and fine-grained classification," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 466–480.

[3] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3973–3981.

[4] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1114–1123.

[5] F. Zhou and Y. Lin, "Fine-grained image classification by exploring bipartite-graph labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1124–1133.

[6] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200-2011 dataset," Dept. Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[7] T. Berg, J. Liu, S. Woo Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur, "Birdsnap: Large-scale fine-grained visual categorization of birds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2011–2018.

[8] S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie, "The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization," *Int. J. Comput. Vis.*, vol. 108, nos. 1–2, pp. 3–29, 2014.

[9] S. Li, K. Li, and Y. Fu, "Self-taught low-rank coding for visual learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 645–656, Mar. 2018.

[10] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. (2013). "Fine-grained visual classification of aircraft." [Online]. Available: https://arxiv.org/abs/1306.5151

[11] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1449–1457.

[12] X. Zhao *et al.*, "Scalable linear visual feature learning via online parallel nonnegative matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2628–2642, Dec. 2016.

[13] L. Niu, W. Li, D. Xu, and J. Cai, "An exemplar-based multi-view domain generalization framework for visual recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 259–272, Feb. 2018.

[14] T. Rumbell, S. L. Denham, and T. Wennekers, "A spiking self-organizing map combining STDP, oscillations, and continuous learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 894–907, May 2014.

[15] C. Zhang, C. Liang, L. Li, J. Liu, Q. Huang, and Q. Tian, "Fine-grained image classification via low-rank sparse coding with general and class-specific codebooks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1550–1559, Jul. 2017.

[16] H. Zhang, L. Cao, and S. Gao, "A locality correlation preserving support vector machine," *Pattern Recognit.*, vol. 47, no. 9, pp. 3168–3178, Sep. 2014.

[17] S. Branson, G. Van Horn, S. Belongie, and P. Perona. (2014). "Bird species categorization using pose normalized deep convolutional nets." [Online]. Available: https://arxiv.org/abs/1406.2952

[18] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNS for fine-grained category detection," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[19] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 842–850.

[20] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, and C.-L. Liu, "LG-CNN: From local parts to global discrimination for fine-grained recognition," *Pattern Recognit.*, vol. 71, pp. 118–131, Nov. 2017.

[21] J. Wang *et al.*, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.

[22] W. Shi, Y. Gong, D. Cheng, X. Tao, and N. Zheng, "Entropy and orthogonality based deep discriminative feature learning for object recognition," *Pattern Recognit.*, vol. 81, pp. 71–80, Sep. 2018.

[23] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.

[24] W. Shi, Y. Gong, J. Wang, and N. Zheng, "Integrating supervised Laplacian objective with CNN for object recognition," in *Proc. Pacific Rim Conf. Multimedia*, 2016, pp. 64–73.

[25] W. Shi, Y. Gong, X. Tao, and N. Zheng, "Training DCNN by combining max-margin, max-correlation objectives, and correntropy loss for multilabel image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2896–2908, Jul. 2018.

[26] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. San Diego, CA, USA, Jun. 2005, pp. 539–546.

[27] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[28] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.

[29] W. Shi, Y. Gong, and J. Wang, "Improving CNN performance with min-max objective," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2004–2010.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2169–2178.

[31] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.

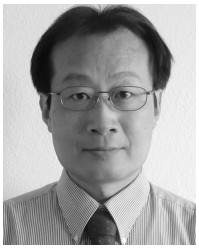
[32] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[33] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 729–736.

[34] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 321–328.
[35] J. Krause, T. Gebru, J. Deng, L.-J. Li, and L. Fei-Fei, "Learning features and parts for fine-grained recognition," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 26–33.
[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[37] W. Shi, Y. Gong, X. Tao, J. Wang, and N. Zheng, "Improving CNN performance accuracies with min–max objective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2872–2885, Jul. 2018.
[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[39] S. Ruder. (2017). "An overview of multi-task learning in deep neural networks." [Online]. Available: https://arxiv.org/abs/1706.05098
[40] F. Chazal, D. Cohen-Steiner, and Q. Mérigot, "Geometric inference for probability measures," *Found. Comput. Math.*, vol. 11, no. 6, pp. 733–751, 2011.
[41] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[42] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[44] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 675–678.

Weiwei Shi received the M.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2012, where he is currently pursuing the Ph.D. degree with the Institute of Artificial Intelligence and Robotics.

His current research interests include image classification, person reidentification, object detection, and multimedia analysis.



Yihong Gong (SM'12–F'17) received the B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from the University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively.

He was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, for four years. From 1996 to 1998, he was a Project Scientist with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. In 1999, he joined NEC Laboratories America, Inc., Princeton, NJ, USA, and established the Media Analytics Group for the labs, where he became the Site Manager to lead the entire Cupertino branch of the labs in 2006. In 2012, he joined Xi'an Jiaotong University, Xi'an, China, where he became a Distinguished Professor of the National Thousand Talents Program, the Vice Director of the National Engineering Laboratory for Visual Information Processing, and the Chief Scientist of the China National Key Basic Research Project (973 Project). His current research interests include pattern recognition, machine learning, and multimedia content analysis.

Xiaoyu Tao received the B.S. degree in software engineering from Xi'an Jiaotong University, Xi'an, China, in 2014, where he is currently pursuing the Ph.D. degree in pattern recognition with the Institute of Artificial Intelligence and Robotics.

His current research interests include image classification, object detection, and face recognition.

De Cheng received the B.S. and Ph.D. degrees in automation control from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2011 and 2017, respectively.

From 2015 to 2017, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Lecturer with XJTU. His research interests include pattern recognition, machine learning, and multimedia analysis.

Nanning Zheng (SM'93–F'06) received the B.S. degree from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975, the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981, and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985.

In 1975, he joined Xian Jiaotong University, where he is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics. His current research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems.

Dr. Zheng became a member of the Chinese Academy of Engineering in 1999. He is the Chinese Representative on the Governing Board of the International Association for Pattern Recognition. He also serves as an Executive Deputy Editor of the Chinese Science Bulletin.