Pattern Recognition 80 (2018) 118–128

Contents lists available at ScienceDirect

Pattern Recognition

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

Deep super-class learning for long-tail distributed image classification

Yucan Zhou, Qinghua Hu*, Yu Wang

School of Computer Science and Technology, Tianjin University, China

ARTICLE INFO

Article history:
Received 29 June 2017
Revised 6 February 2018
Accepted 4 March 2018
Available online 7 March 2018

Keywords:
Super-class construction
Block-structured sparsity
Deep learning
Long-tail distribution

ABSTRACT

Long-tail distribution is widespread in many practical applications, where most categories contain only a small number of samples. As sufficient instances cannot be obtained for describing the intra-class diversity of the minority classes, the separating hyperplanes learned by traditional machine learning methods are usually heavily skewed. Resampling techniques and cost-sensitive algorithms have been introduced to enhance the statistical power of the minority classes, but they cannot infer more reliable class boundaries beyond the description of samples in the training set. To address this issue, we cluster the original categories into super-class to produce a relatively balanced distribution in the super-class space. Moreover, the knowledge shared among categories belonging to a certain super-class can facilitate the generalization of the minority classes. However, existing super-class construction methods have some inherent disadvantages. Specifically, taxonomy-based methods suffer a gap between the semantic space and the feature space, and the performance of learning-based algorithms strongly depends on the features and data distribution. In this paper, we propose a deep super-class learning (DSCL) model to tackle the problem of long-tail distributed image classification. Motivated by the observation that classes belonging to the same super-class usually have more similar evaluations on the features than those belonging to different super-classes, we design a block-structured sparse constraint and attach it on the top of a convolutional neural network. Thus, the proposed DSCL model can accomplish representation learning, classifier training, and super-class construction in a unified end-to-end learning procedure. We compared the proposed model with several super-class construction methods on two public image datasets. Experimental results show that the super-class construction strategy is effective for the long-tail distributed classification, and the DSCL model can achieve better results than the other methods.

1. Introduction

Long-tail distribution learning is a special classification task, where more than hundreds of labels should be learned, and different categories of samples are long-tail distributed, such as Oxford 102 Flowers Dataset [1] and SUN 397 Scene Categorization Dataset [2]. In fact, the long-tail distribution widely exists in various real-world applications, such as object detection [3–5], scene parsing [6–8], document processing [9–11], and item recommendation [12,13].

It is challenging to model the long-tail distribution well for traditional machine learning methods, and even for deep models which have achieved state-of-the-art performance in various tasks [14]. There are three main challenges. Firstly, the minority classes in the tail jointly make up an important portion of the entire dataset. Therefore, they are very significant and cannot be excluded merely to obtain a uniform distribution. Secondly, the weak statistical ability of the minority classes makes the training loss dominated by the majority classes, resulting in the separating hyperplane heavily skewed to the minority classes. Thirdly, samples of the minority class are incapable of describing its intra-class diversity. So the class boundary is condensed heavily around these samples in the training set. Thus, an instance with different features from samples of the same category in the training set cannot be recognized correctly.

Some imbalance learning techniques can be introduced to solve the long-tail distribution problem. In general, these methods can be divided into two categories: the data-preprocessing techniques and the algorithmic approaches. The data-preprocessing techniques include under-sampling [15], up-sampling [16], and synthetic data generation [17]. These methods attempt to generate balanced datasets. However, the under-sampling method may lose valuable samples in the majority classes, and the up-sampling method and the synthetic data generation method increase the computational cost. In an algorithmic approach, the minority class is regarded as the anomaly, then, one-class learning methods are applied to rec-

* Corresponding author. Tel.: +862227401839, fax: +862227401839.
E-mail addresses: zhouyucan@tju.edu.cn (Y. Zhou), huqinghua@tju.edu.cn (Q. Hu), armstrong_wangyu@tju.edu.cn (Y. Wang).

https://doi.org/10.1016/j.patcog.2018.03.003
0031-3203/

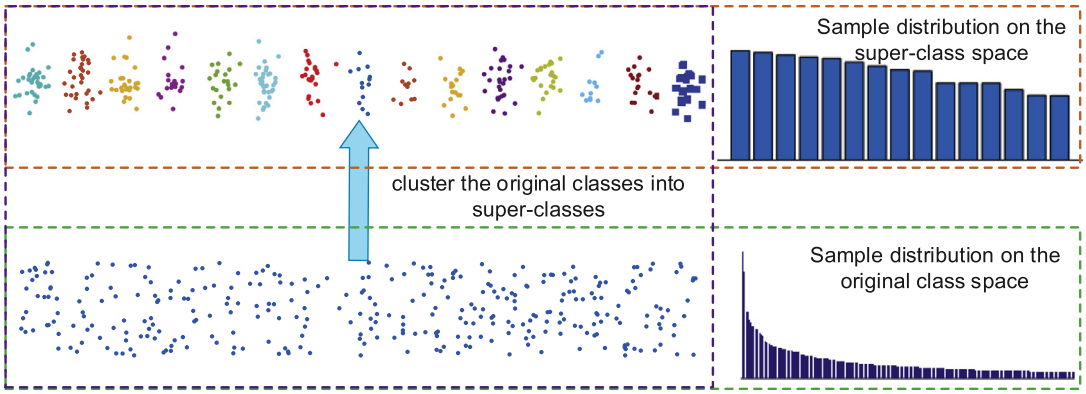

Fig. 1. Example of balanced sample distribution in the super-class space. Each dot in the figure represents a category.

ognize samples belonging to that class [18–20]. As there are many minority classes in the long-tail distribution learning, transferring to the anomaly detection is not applicable. Cost-sensitive learning is another algorithmic approach dealing with the imbalanced classification. It aims to shift the bias of a classifier to favor the minority class by maximizing a weighted loss function defined by a cost matrix based on the dataset [21,22]. Since the minority class is favorable, there is a high probability that the classifier will misclassify some samples of the majority class. Some new loss functions [23,24] have also been designed on the sample pairs or triplets to reduce the impact of imbalance. Yuan et al. propose a regularized ensemble framework for imbalanced training data, which penalizes the classifier when it misclassifies examples that were correctly classified in the previous learning phase [25].

Neither the data-preprocessing techniques nor the algorithmic approaches can guarantee an improved performance on the unseen data of the minority class. Moreover, a flat scheme is usually adopted to transfer the binary classification to the multi-class classification, i.e., one-vs.-all or one-vs.-one classifiers. For a dataset containing L categories, the one-vs.-all strategy needs to compute L classifiers, and the one-vs.-one strategy requires $L(L-1)/2$ computations, when predicting a sample in the testing phase. When L is large, prediction using either of these two flat methods is time-consuming.

To address these issues, we cluster categories into different super-classes. The effectiveness of this strategy for large-scale classification has been verified in some recent papers [14,26]. It has three main advantages. Firstly, when the original classes have been clustered into some super-classes, distribution of the samples in the super-class space can be relatively balanced (as shown in Fig. 1). Secondly, classes belonging to the same super-class can share their knowledge, which helps the minority classes generalize well [27]. Fig. 2 presents an example that illustrates the benefit of the shared knowledge among categories in a super-class. When the training set contains only a few images and cannot describe the varieties of the petunia (containing only light-colored samples), it is almost impossible to recognize a dark purple petunia. However, given the knowledge that the petunia is similar to the mirabilis in the shape of its flowers, comparable to the glory in its stamens, and similar to the euphorbia milii in the form of its leaves, the dark purple petunia can meet these criteria and be probably recognized. Thirdly, for the tree-like label structure (from the super-class to the original class), a hierarchical classifier can be applied. Thus, the prediction time can be reduced to $O(log_K L)$, where K is the number of super-class. Moreover, the searching space (i.e., the label space) is compressed for each classifier in the hierarchy.

To automatically learn an effective super-class structure, a deep super-class learning (DSCL) model is proposed in this paper. Fig. 3 shows an example of a dataset containing images of birds and flowers. In the bird set, the color of the feathers, shape and length of the tail, the beak, and the shape of the claw are relatively important for distinguishing different kinds of birds. In the flower set, however, features such as the color and number of the petals, shape of the leaves, and shape and size of the stamen are more discriminative than others. Thus, we can conclude that different clusters of classes (i.e., the super-classes) value different subsets of features. Based on this observation, a block-structured sparse regularization term on the weight matrix of the classification layer is attached to the objective function of a convolutional neural network (CNN). This regularization term is a trade-off between the common feature space selection and the characteristic preservation, which will make the weight matrix aggregated into different sets. Therefore, the proposed model can accomplish the tasks of representation learning, classifier training, and label structure extraction simultaneously.

The main contributions of this paper are summarized as follows.

- We propose a DSCL model for long-tail distribution classification. A block-structured sparse regularization term is designed and attached to the objective function. Thus, the deep model can obtain the super-class structure while learning the features and the classifier in an end-to-end procedure.
- The weight matrix of the classification layer learned by the proposed model indicates the different importance evaluations on the learned representation, which implies the cluster structure of the original classes.
- We present the performance evaluation of the proposed model on two real-world image datasets. The experimental results demonstrate that the super-class construction strategy can achieve better results for the long-tail distribution classification, and the super-class structure learned with the DSCL model can further improve the performance.

The rest of this paper is organized as follows. In Section 2, related work is introduced. In Section 3, the proposed DSCL model and the corresponding algorithm are described in detail. Experimental results and analysis are provided in Section 4. Finally, we conclude this paper and discuss future work in Section 5.

2. Related Work

2.1. Super-class learning

In general, there are two strategies to construct the super-class structure for the original classes.

The first strategy depends on a given high-level semantic structure (e.g., WordNet), which is commonly used to build the label hi-

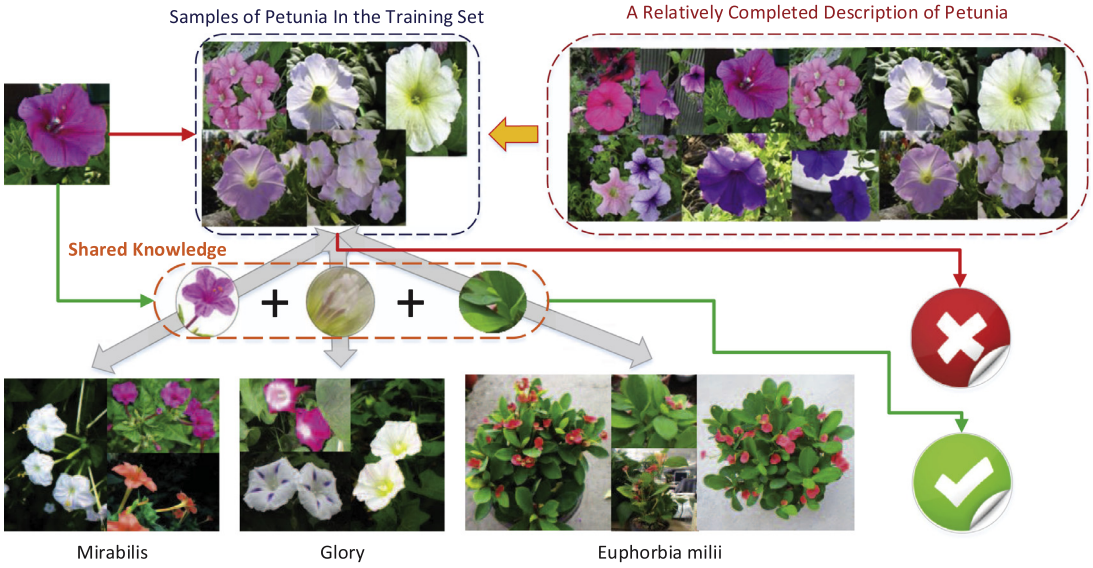

Fig. 2. Illustration of the shared knowledge.

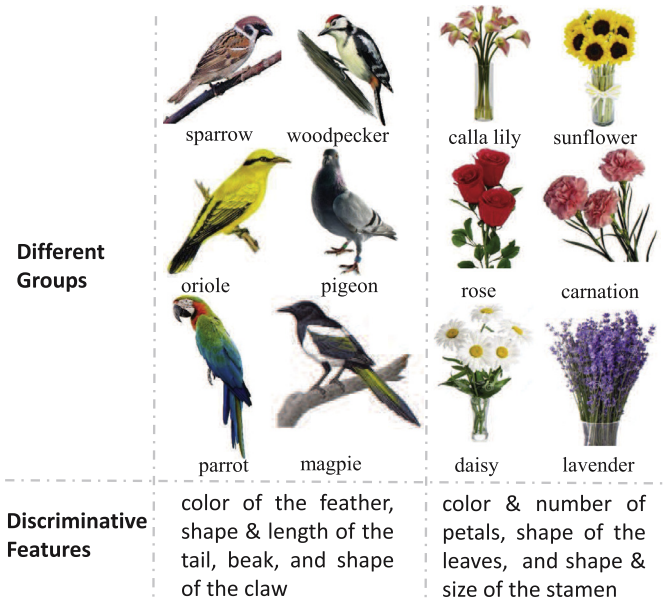

Fig. 3. Illustration of the fact that categories from the same species will attach importance to a more similar feature subset than those from different groups.

erarchy to obtain a better performance [28–30]. However, extracting the super-class structure from the large-scale WordNet is time-consuming, especially when there is quite a large number of categories. In addition, the knowledge of semantic relations is usually not available in practical applications. It is noteworthy that there is a very wide gap between the semantic structure and the visual features, which may confuse the training of classifiers. For example, "whale" and "shark" have similar visual features, but they belong to different semantic clusters, i.e., mammal and fish [31]. Sun et al. show that the classification accuracy decreases when the semantic structure is introduced into training hierarchical classifiers [32].

The second strategy learns the super-class structure automatically based on certain criteria. An intuitional approach for constructing the label structure is to use the confusion matrix of a certain classifier. In a confusion matrix, elements in the main diagonal show the numbers of the samples that are correctly classified, and the others are the numbers of the samples that the classifier cannot distinguish, indicating the differentiation of the two classes. Some researchers [33,34] construct the label structure by assigning the confused classes to one set. However, the calculation of the confusion matrix is usually time-consuming. More importantly, the confusion matrix may not be reliable because of the heavily imbalanced data distribution.

To reduce the impact of the data distribution, the affinity matrix was introduced. Zhou et al. use the mean Euclidean distance of all the sample-pairs of any two categories to define the affinities of different categories [35]. Dong et al. compute it by averaging the kernel distances of all the instances from each two categories [36]. In these methods, it takes $O(L^2N^2)$ calculations to obtain the affinity matrix, where L is the number of the labels and N is the number of samples in the dataset. Qu et al. reduced the computational complexity to $O(LN)$ by calculating the affinity matrix with certain statistical values rather than the pairwise distances [37]. Although the similarity-based methods have shown improvements in accuracy, their performance relies heavily on the feature representation. Generally, the learning-based methods consist of two steps: when a confusion matrix or an affinity matrix has been calculated, a clustering method [35,38,39] is applied to obtain the super-class structure. Therefore, the construction procedure of the super-class structure is separated into classifier training and feature learning, making it nonadjustable.

2.2. Deep learning

In recent years, deep learning has achieved noteworthy accuracy in many complicated real-word applications, including image recognition [40], speech processing [41], video analysis [42], and others [43,44], when a large amount of training data is available. The main contribution of deep learning is that it simplifies the learning paradigm of many complicated tasks into an end-to-end procedure. For object classification, speech recognition, and natural language processing, instead of the carefully hand-crafted features, raw data is required by deep models [40,45]. By slicing one output layer into multiple output layers, deep models can perform multi-task learning [46]. Moreover, deep models do transfer learning by transferring parameters of the source to the target [47]. When training data, architecture, and loss function are available,

feature extraction and classification can be accomplished simultaneously through the back-propagation algorithm.

To avoid overfitting and improve performance on limited datasets, sparse assumptions on the connectivity (i.e., the weight matrix) are introduced. The ℓ_2 norm minimizes the quadratic sum of each element in the weight matrix [48–50] during the training, while the ℓ_1 norm restricts the sum of absolute values of the weights [51,52]. Zou and Hastie [53] and Kang et al. [54] suggest simultaneously imposing the ℓ_1 norm and ℓ_2 norm. Scardapane et al. [55] and Zhao et al. [56] introduce group sparse regularization to accomplish feature selection while training a deep classifier. Yuan et al. designed a manifold regularizer to exploit the structural semantic information between images [57]. Our proposed model is different from these previous methods. Instead of a certain sparse assumption on the whole weight matrix of a deep neural network (DNN), we designed a block-structured sparse constraint only for the connectivity to the classifier layer. Then, this connectivity is used to obtain the super-classes.

3. Proposed model

We now present the deep super-class learning model for long-tail distribution classification. We first provide basic knowledge and notations of deep learning. In Section 3.1, we describe the architecture of the proposed DSCL model and the principle for learning the super-class structure with this model. Then, the objective function of DSCL, especially the regularization term, is introduced in detail in Section 3.2. In the following subsection, we focus on inferring the derivative, which is required by the batch gradient descent optimization algorithm (Section 3.3).

A general DNN can be denoted by the function $y = f(x; \mathbf{W})$. It propagates an input vector $x \in R^D$ through H hidden layers to obtain the output vector $y \in R^L$. $\mathbf{W}$ represents all the adaptable parameters of the network, where $1 \le k \le H+1$. For the kth hidden layer, operations on an input vector $\mathbf{h}_k$ to return the output vector $\mathbf{h}_{k+1}$ can be defined as:

$$\mathbf{h}_{k+1} = g_k(\mathbf{W}_k \mathbf{h}_k + \mathbf{b}_k), \tag{1}$$

where $\{\mathbf{W}_k;\ \mathbf{b}_k\}$ represents the adaptable parameters of this layer, and $g_k(\cdot)$ is an activation function to be applied element-wise. Given a specific training set containing N samples $\{(x_1, l_1), (x_2, l_2), \ldots, (x_N, l_N)\}$ of L classes, where each input x_i is labeled as class l_i, the network is optimized by minimizing a standard regularized cost function:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \left\{ \frac{1}{N} \sum_{i=1}^{N} L(l_i, f(x_i, y)) + \lambda R(\mathbf{w}) \right\}, \tag{2}$$

where $L(\cdot,\ \cdot)$ is a proper cost function, $R(\cdot)$ is the regularization term, which reflects certain prior knowledge, and the scalar coefficient $\lambda \in (0, 1)$ weights the importance of these two terms.

3.1. Framework of deep super-class learning

The proposed method can be readily applied to other network architectures. Here, we give an implementation on the VGG-S [58], a typical CNN. Fig. 4 shows the architecture of the DSCL model for large-scale image classification with long-tail distribution. The target is to learn the high-level abstract representations, a deep and powerful classifier, and label structure in an end-to-end learning procedure. Given a batch of training images and their label (x, y), we propagate them through the network until they reach the output layer, and compute the loss and gradient. Then, the derivative of the loss is back-propagated to each layer. After adding the derivative of the regularization term for parameters in every layer, the parameters of the entire network are updated. Usually, the weight parameters of all layers share the same regularization formula.

In this paper, we introduce a new type of regularization on the weight parameters of the last fully connected layer. This new regularization term contains a column sparsity constraint and an item-wise constraint, which operate together to group the weight parameters into different clusters. Then, the label structure can be inferred from this weight matrix. Here, we denote the weight matrix of the classification layer by $\mathbf{W}_c$, and the other weight parameters by $\mathbf{W}_o$.

3.2. Deep super-class learning

As mentioned, the parameters of the network are optimized by minimizing a loss function and a regularization term (as shown in Formula (3)) on the training dataset.

$$E(x, l) = \frac{1}{N} \sum_{i=1}^{N} L(l_i, f(x_i, y)) + \lambda R(\mathbf{W}). \tag{3}$$

Traditionally, softmax with loss is employed in a CNN for image categorization with high accuracy. Given the output of the last classification layer $f(x,\ y) \in R^L$, where each item represents the probability of the input x belonging to each category, the loss of softmax can be defined as the negative log-likelihood on $(x,\ l)$:

$$f_s(x, l) = -log \underbrace{\frac{e^{f(x,l)}}{\sum_{l'} e^{f(x,l')}}}_{P(l|x)}, \tag{4}$$

where $P(l|x)$ is the posterior probability of the input x being classified as the lth class.

In fact, this loss function attempts to squeeze training data from the same class into a corner of the feature space to obtain the minimum loss. Thus, the intra-class variance cannot be preserved, which may lead to overfitting on the training data. To solve this problem, the regularization strategy is recommended. The squared l_2 norm and the l_1 formulation are the two most common constraints on the weights, which are defined as Formulas (5) and (6), respectively. In this study, we used the l_2 norm for regularizing $\mathbf{W}_o$.

$$R_{l_2}(\mathbf{W}) = \|\mathbf{W}\|_2^2 = \sum_{i,j} w_{i,j}^2, \tag{5}$$

$$R_{l_1}(\mathbf{W}) = \|\mathbf{W}\|_1 = \sum_{i,j} \left| w_{i,j} \right|. \tag{6}$$

The structure of the weight parameters $\mathbf{W}_c$ implies the super-class information of the original labels, which is the main primary target. Motivated by the observation that some categories are strongly related, since they give relatively similar values to the features, and the fact that $\mathbf{W}_c$ reflects the feature evaluation of each category, we propose a block-structured sparse regularization term for $\mathbf{W}_c$.

$$R(\mathbf{W_c}) = \alpha R_{l_{cs}}(\mathbf{W}_c) + (1-\alpha) R_{l_{1,1}}(\mathbf{W}_c), \tag{7}$$

where

$$\begin{aligned} R_{l_{cs}}(\mathbf{W}) &= \|\mathbf{W}^T\|_{2,1} = \sum_i \| \sum_j w_{i,j}^2 \|, \\ R_{l_{1,1}}(\mathbf{W}) &= \sum_l \|\mathbf{W}_l\|_1 = \sum_i \| \sum_j \left| w_{i,j} \right| \|_1 \\ &= \sum_{i,j} \left| w_{i,j} \right| = R_{l_1}. \end{aligned} \tag{8}$$

In Formula (7), there are two regularization terms. The first, $R_{l_{cs}}(\mathbf{W}_c)$, encourages all labels to fall into one cluster by eliminating the shared features that are not valuable for all the categories,

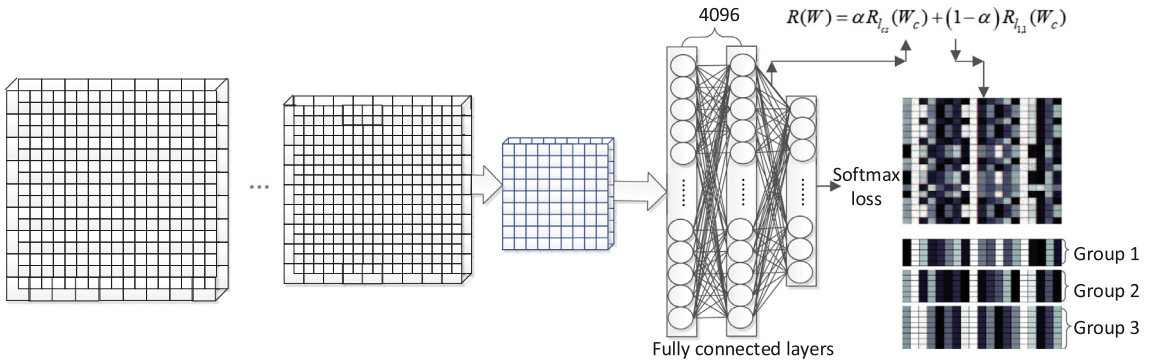

Fig. 4. Architecture of the proposed model for long-tail distribution classification.

and the second, $R_{l_{1,1}}(\mathbf{W}_c)$, reflects an element-wise sparse structure, and prompts every class to be unique and to choose features that are of use only for itself. In summary, the penalties of $R_{l_{cs}}(\mathbf{W}_c)$ and $R_{l_{1,1}}(\mathbf{W}_c)$ operate together to choose different sets of features for different clusters, and thus the super-class structure can be discovered, while the parameter α balances the importance of these two regularization terms.

Finally, our objective function to train the CNN is

$$\min_{\mathbf{W}_o,\mathbf{W}_c} \frac{1}{N}\sum_{i=1}^{N} -log\frac{e^{f(x_i,l_i)}}{\sum_{y_i=1}^{L} e^{f(x_i,y_i)}} + \lambda\|\mathbf{W}_o\|_2^2 + \gamma(\alpha\|\mathbf{W}_c^T\|_{2,1} + (1-\alpha)\|\mathbf{W}_c\|_1), \tag{9}$$

where $\lambda \in [0, 1)$, $\gamma \in [0, 1)$, and $\alpha \in [0, 1)$.

3.3. Optimization

We optimize the DSCL model by using a batch gradient descent algorithm. The core of a gradient-based optimization algorithm is the computation of the first derivative of the objective function. So that the development of the algorithm is convenient, we rewrite Formula (9) in the form

$$\min_{\mathbf{W}_o,\mathbf{W}_c} \underbrace{\frac{1}{N}\sum_{i=1}^{N} -log\frac{e^{f(x_i,l_i)}}{\sum_{y_i=1}^{L} e^{f(x_i,y_i)}}}_{J_1} + \underbrace{\lambda\|\mathbf{W}_o\|_2^2}_{J_2} + \underbrace{\alpha\|\mathbf{W}_c^T\|_{2,1} + \beta\|\mathbf{W}_c\|_1}_{J_3}, \tag{10}$$

where $\lambda \in [0, 1)$, $\alpha \in [0, 1)$, and $\beta \in [0, 1)$.

For $\mathbf{W}_c$, the gradient can be computed as

$$\begin{aligned}\nabla_{\mathbf{W}_c}(i,j) &= \frac{\partial J_1}{\partial \mathbf{W}_c(i,j)} + \frac{\partial J_3}{\partial \mathbf{W}_c(i,j)} \\ &= (P(y_i|x) - \mathbf{1}_{[y_i=l]})\nabla_{g_i} h_H(j) \\ &\quad + \alpha\frac{\mathbf{W}_c(i,j)}{\sum_{j=1}^{n_H}\mathbf{W}_c(i,j)^2} \\ &\quad + \beta sign(\mathbf{W}_c(i,j)),\end{aligned} \tag{11}$$

where $P(y_i|x)$ is calculated with Formula (4), and n_H is the number of neurons in the Hth layer. With the chain rule, the derivative of the kth weight matrix, $k \in [1, H]$, is

$$\begin{aligned}\nabla_{\mathbf{W}_k}(i,j) &= \sum_{m=1}^{n_{k+2}} \mathbf{W}_{k+1}(m,i)\nabla_{g_i}\mathbf{h}_k(j) \\ &\quad + 2\lambda\mathbf{W}_k(i,j).\end{aligned} \tag{12}$$

The parameters of the DSCL model are initialized with those of a pre-trained VGG-S net. Then we fine tune the weights with Formulas (11) and (12).

4. Experiments

In this section, we introduce the experiments that were conducted on two real-world image datasets, i.e., Oxford (Oxford 102 Flowers dataset) [1] and SUN (SUN 397 Scene Categorization dataset) [2], in which most categories consists of only a few samples. To show the effectiveness of the label structure learned by the DSCL model, we compare it with the semantic structure and two additional structures constructed by applying clustering algorithms to the confusion matrix and affinity matrix. The experimental results demonstrate that the super-class-based strategy can achieve a consistently inspiring improvement for long-tail distribution learning, and our method performs better than the other methods on both Oxford and SUN.

In this study, we implement the proposed DSCL model with Caffe. To achieve our goal, we design an inner_product_block_sparse_layer by adding the block-structured sparse constraint to its weight matrix. Then, we use this inner_product_block_sparse_layer instead of the original inner_product_layer as the output layer of the VGG-S. The weight_decay parameter of the output layer is set to zero. Therefore, only the block-structured sparse constraint is active in this layer.

4.1. Datasets

We conducted our experiments on the Oxford and SUN datasets to show the performance of different methods, all of which have a long-tail data distribution, as shown in Fig. 5. Moreover, a semantic structure is attached to the SUN dataset. For both the datasets, we randomly selected 75% of all the images as the training set and the remaining samples formed the testing set. In our experiments, the dataset partitions were exactly the same for our approach and all the compared methods to allow a fair comparison. The two image datasets are now briefly described as follows.

Oxford 102 Flowers Dataset. This dataset was collected by the Visual Geometry Group in Oxford. It consists of 102 flower categories that are common in the United Kingdom. In total, it contains 8,189 images, and the number of images in each class varies from 40 to 258. The owner offers a partition of the training, testing, and validation set with uniform data distribution in the training set. Therefore, we split it into a training and testing set again randomly to ensure the sets share the same data distribution as the entire dataset.

SUN 397 Scene Categorization Dataset. This dataset is a well-sampled subset of the Scene Understanding (SUN) dataset with 397 different scenes. The owners manually built an over-completed two-level taxonomy for the scene categories. The 397 classes are classified to 15 more general sets at the second level (basic-level categories) that are in turn connected to 3 nodes at the first level

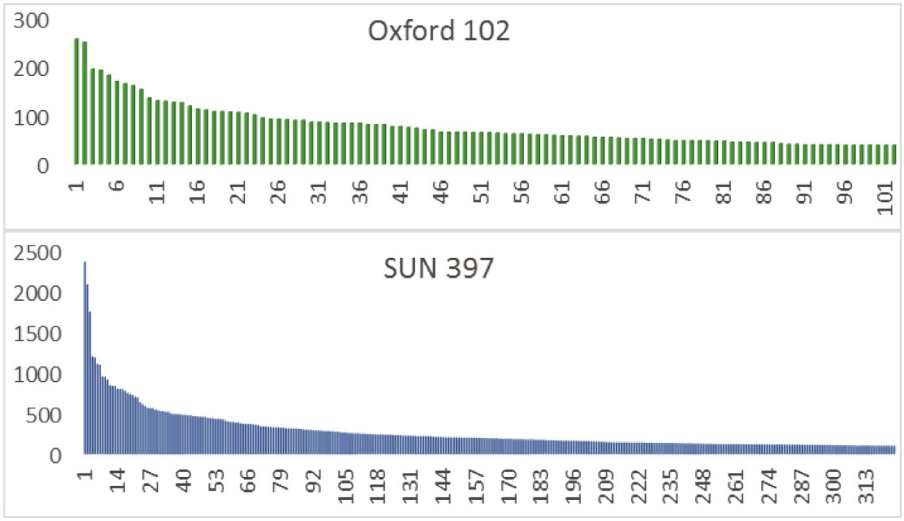

Fig. 5. Sample distribution of Oxford and SUN.

Table 1
Classification results on the label level.

Dataset	Method	Accuracy	mAP
Oxford	Flat	91.597%	90.432%
	ST-Ex	–	–
	AM-SP	87.543%	86.619%
	CM-SP	87.885%	86.781%
	DSCL	**95.164%**	**94.683%**
SUN	Flat	66.884%	61.668%
	ST-Ex	66.616%	61.962%
	AM-SP	67.295%	62.329%
	CM-SP	67.388%	62.793%
	DSCL	**70.895%**	**65.876%**

Table 2
Classification results on the super-class level.

Dataset	Method	Accuracy	mAP
Oxford	Flat	91.597%	90.432%
	ST-Ex	–	–
	AM-SP	89.692%	87.275%
	CM-SP	89.741%	86.670%
	DSCL	**97.997%**	**94.212%**
SUN	Flat	66.884%	61.668%
	ST-Ex	80.254%	**79.514%**
	AM-SP	**83.703%**	78.488%
	CM-SP	83.667%	78.598%
	DSCL	83.237%	67.707%

(superordinate categories). However, this hierarchy is a directed acyclic graph with 73 scenes belonging to more than one basic-level category. For simplicity, we exclude these categories. Thus, the SUN dataset finally had 324 classes and 90,261 images in total. The number of images varied across categories, but there were at least 100 images per category. The basic-level categories were adopted to construct the semantic label structure.

4.2. Comparison methods and evaluation metrics

We compared the super-class structure learned by the DSCL model with that extracted from the semantic taxonomy, and another two structures, trained by two state-of-the-art algorithms, provided by the raw data. For the learning-based methods, the spectral clustering algorithm was applied on a confusion matrix and an affinity matrix. To make the confusion matrix and the affinity matrix more reliable, we used the features extracted from a pre-trained VGG-s net.

Table 3
Classification results on the super-class level.

Dataset	Method	The label level		The super-class level	
		Accuracy	mAP	Accuracy	mAP
Oxford	Flat	91.597%	90.432%	91.597%	90.432%
	AlexNet	95.652%	94.934%	99.609%	87.004%
	VGG-16	95.896%	95.394%	99.609%	89.108%
SUN	Flat	66.884%	61.668%	66.884%	61.668%
	AlexNet	71.644%	66.776%	84.261%	66.810%
	VGG-16	71.773%	67.838%	84.324%	67.719%

ST-Ex: ST-Ex is a method of clustering classes via mining their semantic relationships defined by the WordNet or other domain knowledge. This method has been frequently used to obtain label structures to enhance the performance of many tasks. For the SUN dataset, there exists a manually annotated hierarchy of the classes, and therefore, we clustered categories with the information of the second layer, as mentioned in Section 3.1, whereas for the Oxford dataset, we could not achieve the semantic structure of the categories because we lacked a priori knowledge about the relevancy of different flowers.

AM-SP: AM-SP refers to the method presented in [37], which was proposed to simplify the similarity calculation for every category pair. Then, a hierarchical spectral clustering is applied on the affinity matrix to build the hierarchical inter-class structure. In this study, a super-class structure was required, and therefore, we used the classical spectral clustering algorithm instead of a hierarchical one.

CM-SP: CM-SP learns the label structure with the spectral clustering and the confusion matrix. First, a logistic regression classifier is trained to obtain the confusion matrix, and then, this confusion matrix is delivered to the spectral clustering algorithm to learn the label structure.

Because of the long-tail data distribution, the frequently-used accuracy indicator was not sufficient to evaluate the results of the different methods, and therefore, mAP wa employed as an additional.

4.3. Results and analysis

For DSCL, the label structure is implied in the weight matrix of the last fully connected layer. Thus, the explicit label structure can be modeled explicitly by clustering the weights of all the output nodes into G clusters. In this method, we use d the k-means al-

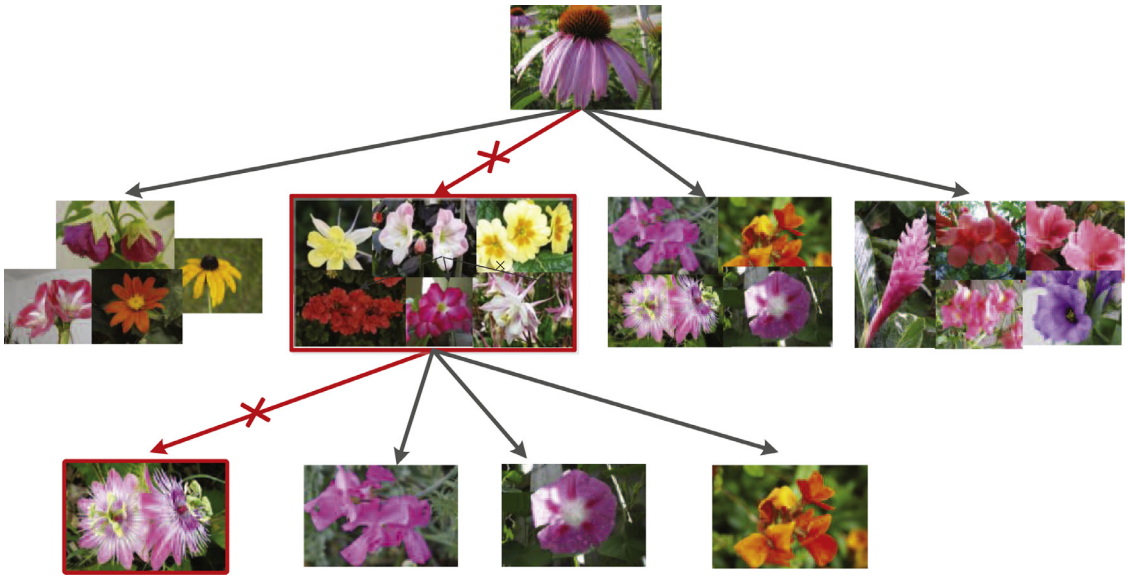

Fig. 6. Classification results of an example with the label structure learned by the AM-SP on Oxford.

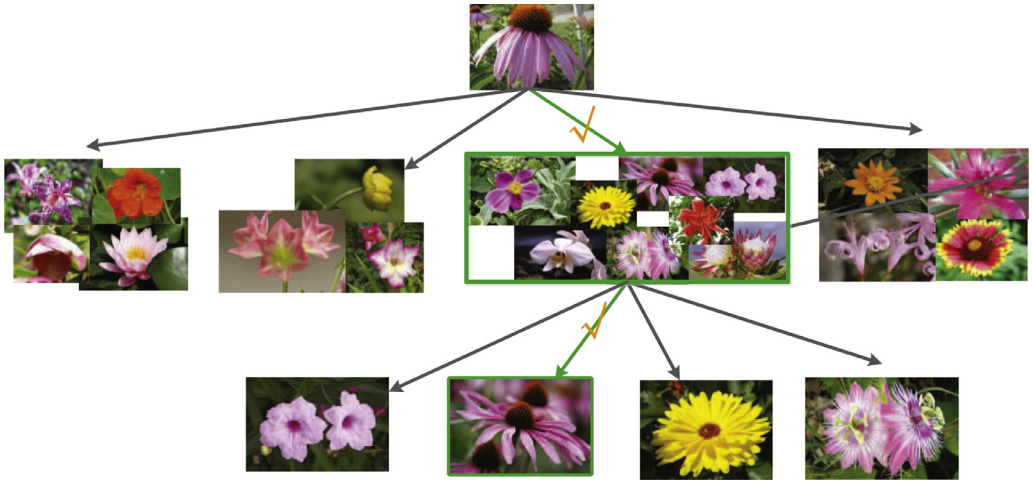

Fig. 7. Classification results of an example with the label structure learned by the DSCL on Oxford.

gorithm to perform the clustering and chose G from the same set of other methods for efficient comparative analysis. As for Formula (10), we set $\lambda = 0.05$ experimentally, and γ was carefully selected from [5, 0.5, 0.05, 0.005, 0.0005].

To fairly assess these different structures, we adopt the top-down strategy, which trains a series of classifiers layer by layer to model the label structure. Here, logistic regression is selected as the base classifier. Moreover, since the number of super-class for the semantic method is fixed, but for the learning-based methods, it can be any value, we chose the target number of the clusters in our experiments from a given set in which the candidates are comparable to that of the semantic method. The SUN dataset is composed of 15 basic-level categories, and the Oxford dataset contains no taxonomy annotation. Thus, the optimal number of clusters is explored in the set [5, 10, 15, 20, 25, 30]. The flat method, which evaluates the performance on the multi-classes classification task, served as the baseline method.

Results on the label level. Table 1 shows the results in terms of the measurements accuracy and mAP on the label level. Here, the top-down logistic regression is applied to the label structures obtained by all the included methods. We can see that the super-class-based methods perform consistently better than the flat method on the two datasets. Specifically, DSCL achieves the best results on both Oxford and SUN. More importantly, the super-class-based methods produce little difference in terms of either accuracy or mAP, which shows their effectiveness in dealing with the long-tail distribution.

Results on the super-class level. The classification results of the different methods on the super-class level are displayed in Table 2. As compared with the flat method, the super-class-based methods obtain obvious improvements when the experiments are conducted on the super-class level. Therefore, when the classifiers are hesitant in deciding the category of a sample, we can provide a super-class label at a high confidence level. This super-class label is significant, since it will simplify the classification task by limiting the candidates to a small subset of labels. For the Oxford dataset, the proposed DSCL method achieves the best result. While on the SUN dataset, the result of DSCL is worse than AM-SP and CM-SP. This is because AM-SP and CM-SP cluster similar categories into a super-class. Thus, the intra-class variation of the super-class is smaller than that of DSCL, which values the discrimination of the fine-grained classes when constructing the super-class structure.

A comparison of Tables 1 and 2 provides some insights. First, the performance of AM-SP and CM-SP is worse on both label and

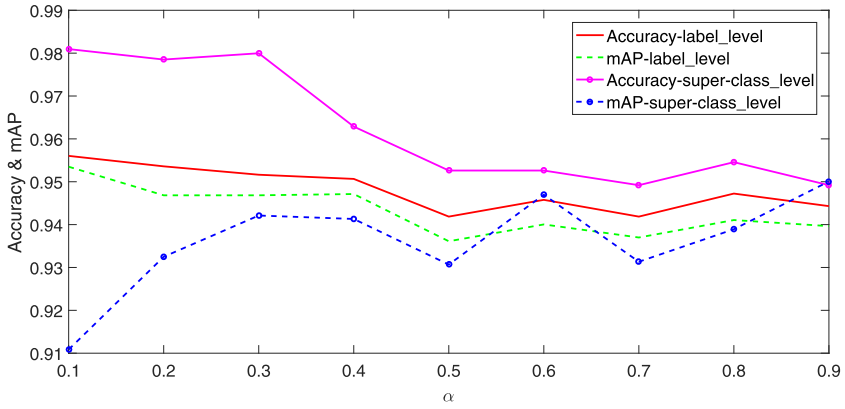

Fig. 8. Classification results with different value of α on Oxford.

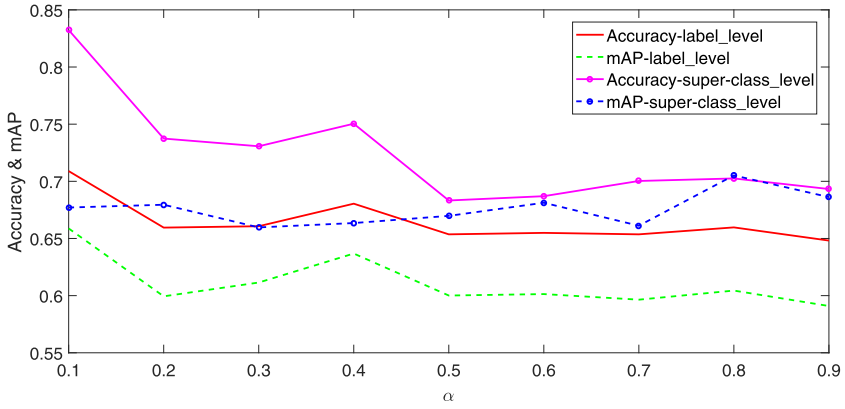

Fig. 9. Classification results with different value of α on SUN.

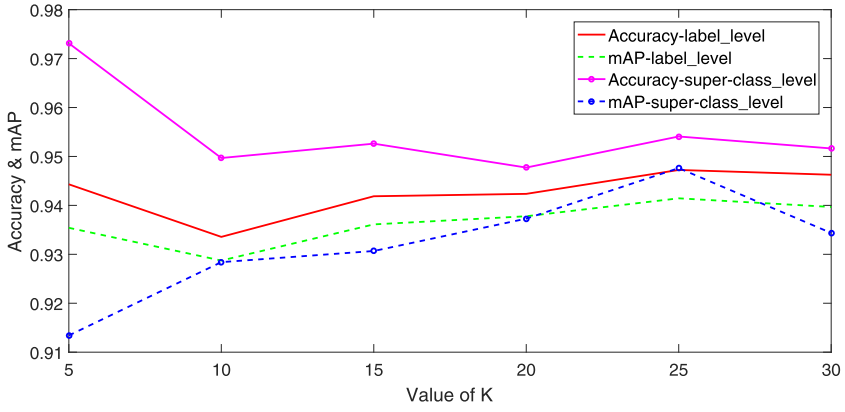

Fig. 10. Classification results with different value of K on Oxford.

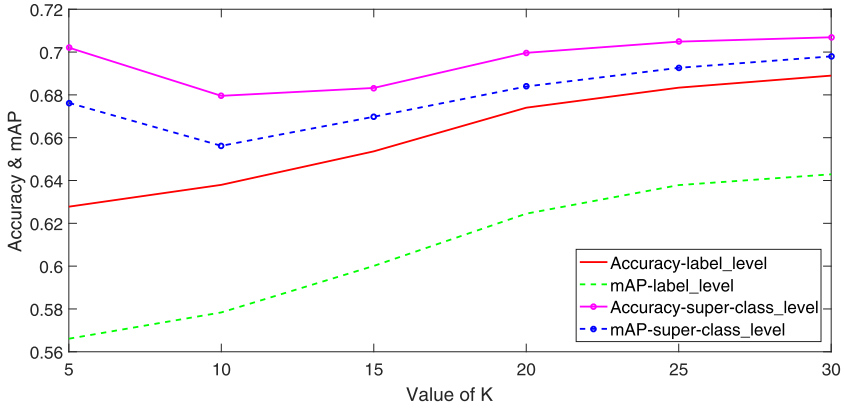

Fig. 11. Classification results with different value of K on SUN.

super-class level on the Oxford dataset. The decline on the label level is probably caused by their poor capabilities on the super-class level. The reasons why they perform poorly may be that they produce a noneffective super-class structure, or that the features contain too many details about the label, which constitutes interfering information and prevents the classifier from classifying samples into the correct super-class. Moreover, the DSCL model achieves approximate results on the two image datasets. The performance of the label level is severely constrained by that of the super-class level. Features for correctly recognizing the super-class of a sample are coarser than those required by label distinction. Therefore, in our opinion these methods would achieve better results if different features were employed for the classifiers on different levels. The performance can be further improved by introducing the features extracted from a CNN fine tuned with the super-class level labels.

Figs. 6 and 7 show the top-down procedure of a sample classified into a certain super-class and class with two different label structures learned by AM-SP and the proposed DSCL. Given an image of a purple coneflower, the top-down classifiers first classify it to a certain super-class, and then a label from this super-class is attached to this image. For the super-class and label level, we present the top 4 candidates. Pictures with bold edges are the candidates with the highest confidence for each level of the two label structures. It is obvious that the correct recognition depends heavily on the results of the super-class level.

4.4. Parameter analysis

Figs. 8 and 9 describe the classification results with different values of α on the Oxford and SUN dataset, when the number of clusters (i.e., K) is set to 25 and 30, respectively. For the super-class level, we can conclude that the accuracy decreases when α gets larger, while the mAP increases. This is because that the intra-class difference become lager when the number of classes belonging to a super-class grows. However, the data distribution on the super-class space are more and more balanced, which will benefit the training of the classifier. For the label level, both the accuracy and mAP decrease, as its performance is severely constrained by that of the super-class level. Considering the accuracy and mAP on both super-class level and label level, we set $\alpha = 0.3$ for Oxford, and $\alpha = 0.4$ for SUN in our experiments.

Figs. 10 and 11 show the classification results with different values of K on the Oxford and SUN dataset, when α is set to 0.3 and 0.4, respectively. For the super-class level, we can see that both the accuracy and mAP increase when K gets larger, which is benefit from the decreased intra-class difference. However, accuracy and mAP decrease at some K-value points. On the Oxford dataset, accuracy drops where $K = 10$ and $K = 30$, while the mAP decreases at the point $K = 10$. On the SUN dataset, The accuracy and mAP on the SUN both get decreased where $K = 10$. As K grows, the solving space of the super-classes becomes larger, making the task more and more difficult. Then, the performance may be reduced. For the label level, both the accuracy and mAP increase, benefiting from results of the super-classes. Considering the accuracy and mAP on both super-class level and label level, we set $K = 25$ for Oxford, and $K = 30$ for SUN in our experiments.

4.5. Implementation on other deep models

Although the designed block-structured sparse constraint is implemented on the VGG-S architecture in this study, it can be easily applied on other deep learning models. Table 3 presents the results on both the label level and the super-class level with implementation on AlexNet and VGG16. To show the effectiveness of the super-class learning strategy, results of the flat model are also listed in the table. It can be concluded that we can achieve better results with a more powerful deep model.

5. Conclusion and future work

In this study, we examined a super-class-based strategy to reduce the impact of long-tail distribution for image classification. As existing super-class learning methods have inherent disadvantages for the long-tail distribution, we proposed a DSCL model. Based on the observation that categories from the same cluster put a more similar value on the features than those from different ones, a block-structured sparse regularization term was designed. By appending this new regularization term on the weight matrix of the classification layer to the original objective function, the DSCL model can execute representation learning, classifier training, and super-class structure construction simultaneously. To show the effectiveness of the super-class construction strategy and the DSCL model, we conducted several experiments on the Oxford dataset and the SUN dataset. The experimental results show that the super-class-based strategy and the proposed model can considerably improve the accuracy and mAP on both the super-class and label level of classification with long-tail distribution.

To the best of our knowledge, very few studies on solving long-tail distribution classification with the super-class-based strategy and deep learning methods exist. Thus, a considerable amount of work is required to further improve the performance of the current methods. In the future, we will explore new strategies to build the label structure, and consider how to model this structure. Moreover, new evaluation criteria that consider the performance of both the super-class and category level are imperative to assess different classifiers for long-tail distribution classification more reasonably.

Acknowledgment

This work is partly supported by National Natural Science Foundation of China under grants 61432011, 61732011, U1435212, and 61502332.

References

[1] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, 2008, pp. 722–729.
[2] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3485–3492.
[3] O. Boiman, E. Shechtman, M. Irani, In defense of nearest-neighbor based image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[4] X. Zhu, D. Anguelov, D. Ramanan, Capturing long-tail distributions of object subcategories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 915–922.
[5] X. Zhu, C. Vondrick, C.C. Fowlkes, D. Ramanan, Do we need more training data? Int. J. Comput. Vis. 119 (1) (2016) 76–92.
[6] J. Yang, B. Price, S. Cohen, M.-H. Yang, Context driven scene parsing with attention to rare classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 3294–3301.
[7] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017.
[8] R. Hong, M. Wang, Y. Gao, D. Tao, X. Li, X. Wu, Image annotation by multiple-instance learning with discriminative feature mapping and selection, IEEE Trans. Cybern. 44 (5) (2014) 669–680.
[9] R. Reinanda, E. Meij, M. de Rijke, Document filtering for long-tail entities, in: Proceedings of the ACM International on Conference on Information and Knowledge Management, ACM, 2016, pp. 771–780.
[10] Z. Hu, H. Qirong, A. Dubey, E. Xing, Large-scale distributed dependent nonparametric trees, in: Proceedings of the International Conference on Machine Learning, ICML, 2015, pp. 1651–1659.
[11] P. Xie, Y. Deng, E. Xing, Diversifying restricted boltzmann machine for document modeling, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1315–1324.

[12] Y.-C. Ho, Y.-T. Chiang, J.Y.-J. Hsu, Who likes it more?: Mining worth-recommending items from long tails by modeling relative preference, in: Proceedings of the ACM International Conference on Web Search and Data Mining, ACM, 2014, pp. 253–262.
[13] S. Wang, M. Gong, H. Li, J. Yang, Multi-objective optimization for long tail recommendation, Knowled. Based Syst. 104 (2016) 145–155.
[14] W. Ouyang, X. Wang, C. Zhang, X. Yang, Factors in finetuning deep model for object detection with long-tail distribution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 864–873.
[15] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory Undersampling for Class-imbalance Learning, IEEE Press, 2009.
[16] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data . 21 (9) (2009) 1263–1284.
[17] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.
[18] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.
[19] B. Raskutti, A. Kowalczyk, Extreme re-balancing for svms: a case study, ACM SIGKDD Explor. Newslett. 6 (1) (2004) 60–69.
[20] N. Japkowicz, Supervised versus unsupervised binary-learning by feedforward neural networks, Mach. Learn. 42 (1) (2001) 97–122.
[21] Y.-A. Chung, H.-T. Lin, S.-W. Yang, Cost-aware pre-training for multiclass cost-sensitive deep learning, Arxiv: 1511.09337 (2015).
[22] S. Feng, C. Lang, J. Feng, T. Wang, J. Luo, Human facial age estimation by cost-sensitive label ranking and trace norm regularization, IEEE Trans. Multimedia 19 (1) (2017) 136–148.
[23] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proceedings of Advances in Neural Information Processing Systems, NIPS, 2014, pp. 1988–1996.
[24] D. Cheng, Y. Gong, S. Zhou, J. Wang, N. Zheng, Person re-identification by multi-channel parts-based cnn with improved triplet loss function, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 1335–1344.
[25] X. Yuan, L. Xie, M. Abouelenien, A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data, Pattern Recognit. 77 (2018) 160–172.
[26] R. Salakhutdinov, A. Torralba, J. Tenenbaum, Learning to share visual appearance for multiclass object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1481–1488.
[27] B. Samy, The battle against the long tail in computer vision, Talk on Workshop on Big Data and Statistical Machine Learning, 2015.
[28] N. Verma, D. Mahajan, S. Sellamanickam, V. Nair, Learning hierarchical similarity metrics, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2280–2287.
[29] V. Ordonez, J. Deng, Y. Choi, A.C. Berg, T.L. Berg, From large scale image categorization to entry-level categories, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2013, pp. 2768–2775.
[30] Z. Kuang, J. Yu, Z. Li, B. Zhang, J. Fan, Integrating multi-level deep learning and concept ontology for large-scale visual recognition, Pattern Recognit. In Press (2018).
[31] S. Zhao, Y. Han, Q. Zou, Q. Hu, Hierarchical support vector machine based structural classification with fused hierarchies, Neurocomputing (2016) 86–92.
[32] M. Sun, W. Huang, S. Savarese, Find the best path: an efficient and accurate classifier for image hierarchies, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2013, pp. 265–272.
[33] G. Griffin, P. Perona, Learning and using taxonomies for fast visual categorization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[34] B. Liu, F. Sadeghi, M. Tappen, O. Shamir, C. Liu, Probabilistic label trees for efficient large scale image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 843–850.
[35] N. Zhou, J. Fan, Jointly learning visually correlated dictionaries for large-scale visual recognition applications, IEEE Trans. Pattern Anal. Mach. Intell. 36 (4) (2014) 715–730.
[36] P. Dong, K. Mei, N. Zheng, H. Lei, J. Fan, Training inter-related classifiers for automatic image classification and annotation, Pattern Recognit. 46 (5) (2013) 1382–1395.
[37] Y. Qu, L. Lin, F. Shen, C. Lu, Y. Wu, Y. Xie, D. Tao, Joint hierarchical category structure learning and large-scale image classification, IEEE Trans. Image Process. 26 (9) (2017) 4331–4346.
[38] H. Lei, K. Mei, N. Zheng, P. Dong, N. Zhou, J. Fan, Learning group-based dictionaries for discriminative image representation, Pattern Recognit. 47 (2) (2014) 899–913.
[39] Y. Zheng, J. Fan, J. Zhang, X. Gao, Hierarchical learning of multi-task sparse metrics for large-scale image classification, Pattern Recognit. 67 (2017) 97–109.
[40] 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, IEEE, 2016, pp. 770–778.
[41] Y. Qian, T. Tan, D. Yu, Neural network based multi-factor aware joint training for robust speech recognition, IEEE/ACM Trans. Audio Speech Lang. Process. 24 (12) (2016) 2231–2240.
[42] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: Deep networks for video classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 4694–4702.
[43] A. Van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in: Proceedings of Advances in Neural Information Processing Systems, NIPS, 2013, pp. 2643–2651.
[44] C. Manning, Understanding human language: Can nlp and deep learning help? in: Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2016. 1–1.
[45] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., Deep speech 2: end-to-end speech recognition in english and mandarin, in: Proceedings of the International Conference On Machine learning, ACM, 2016, pp. 173–182.
[46] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the International Conference On Machine learning, ACM, 2008, pp. 160–167.
[47] M.J. Afridi, A. Ross, E.M. Shapiro, On automated source selection for transfer learning in convolutional neural networks, Pattern Recognit. 73 (2018) 65–75.
[48] R. Gens, P. Domingos, Discriminative learning of sum-product networks, in: Proceedings of Advances in Neural Information Processing Systems, NIPS, 2012, pp. 3239–3247.
[49] N. Wang, D.-Y. Yeung, Learning a deep compact image representation for visual tracking, in: Proceedings of Advances in Neural Information Processing Systems, NIPS, 2013, pp. 809–817.
[50] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Arxiv:1409.1556 (2014).
[51] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks., in: Aistats, 15, 2011, p. 275.
[52] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, Arxiv:1412.6572 (2014).
[53] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. 67 (2) (2005) 301–320.
[54] G. Kang, J. Li, D. Tao, Shakeout: a new regularized deep neural network training scheme, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, 2016.
[55] S. Scardapane, D. Comminiello, A. Hussain, A. Uncini, Group sparse regularization for deep neural networks, Neurocomputing 241 (2017) 81–89.
[56] L. Zhao, Q. Hu, W. Wang, Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso, IEEE Trans. Multimedia 17 (11) (2015) 1936–1948.
[57] Y. Yuan, L. Mou, X. Lu, Scene recognition by manifold regularized deep learning architecture, IEEE Trans. Neural Netw. Learn.Syst. 26 (10) (2015) 2222–2233.
[58] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, (2014). Arxiv:1405.3531.

Yucan Zhou is currently a Ph.D. student with the School of Computer Science and Technology in Tianjin University. Her research interests include artificial intelligence, deep learning, long-tail distribution learning, and hierarchical classification.

Qinghua Hu received B. E., M. E. and Ph.D. degrees from Harbin Institute of Technology, Harbin, China in 1999, 2002 and 2008, respectively. He once worked with Harbin Institute of Technology as assistant professor and associate professor from 2006 to 2011 and a postdoctoral fellow with the Hong Kong Polytechnic University. He is now a full professor with Tianjin University. His research interests are focused on intelligent modeling, data mining, knowledge discovery for classification and regression. He is the PC co-chair of RSCTC 2010, CRSSC 2012, and ICMLC 2014 and severs as referee for a great number of journals and conferences. He has published more than 100 journal and conference papers in the areas of pattern recognition, machine learning and data mining.

Yu Wang received B.E. degree and M.E. degree from Tianjin University, Tianjin, China in 2013 and 2016, respectively. He is currently a Ph.D. candidate with the School of Computer Science and Technology in Tianjin University. His current research interests include hierarchical classification and hierarchical clustering in machine learning with big data and massive categories.