International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-023-01835-4

ORIGINAL ARTICLE

Coarse-to-fine knowledge transfer based long-tailed classification via bilateral-sampling network

Junyan Xu[1,2] · **Wei Zhao**[1,2] · **Hong Zhao**[1,2]

Received: 19 July 2022 / Accepted: 6 April 2023

Abstract
Long-tailed classification faces a considerable challenge from the imbalanced distribution of head and tail data. Re-sampling is a traditional Single Branch Sampling (SBS) method used to adjust data imbalances that effectively improves the performance of long-tailed classification models. Most existing SBS models assume that the classes are independent of each other and ignore the hierarchical relations among the classes. However, the hierarchical structure is exhibited as coarse- and fine-grained semantic relations, a significant knowledge to guide long-tailed classification. In this paper, we propose a Coarse-to-Fine knowledge transfer based Bilateral-Sampling Network (CFBSNet) for long-tailed classification that alleviates the effects of imbalances in long-tailed data and considers coarse- and fine-grained semantic relationships. First, we present a Bilateral-Branch Sampling Network consisting of two sampling branches. The two sampling branches perform reverse sampling and uniform sampling, respectively. Second, we design a Coarse-to-Fine Knowledge Transfer strategy that regulates different learning stages by adjusting loss weight in each task progressively. CFBSNet pays attention to the semantic relationship between tail data and granularity. The experimental results demonstrated the effectiveness of CFBSNet for long-tailed classification tasks. For instance, the classification accuracy of CFBSNet is 3.16% and 2.62% better than that of baseline models on the CIFAR-100-LT and the SUN datasets, respectively.

Keywords Long-tailed classification · Bilateral-branch sampling network · Hierarchical structure · Coarse-to-fine knowledge transfer

1 Introduction

Long-tailed distribution brings many challenges to long-tailed classification in machine learning [1]. Real-world data are usually unbalanced and follow a long-tailed distribution [2]. Figure 1 shows the distribution of long-tailed data. It illustrates how a few classes (head classes) contain most samples while many classes (tail classes) have very few samples. Long-tailed data naturally appear in many applications, such as biomedicine [3], endangered species recognition [4], facial recognition [5], and disaster prediction [6].

Existing methods for solving long-tailed classification problems can be categorized into three main classes: algorithm-level strategies, transfer learning, and data-level strategies. Algorithm-level strategies [7–10] mainly adjust the weight of each class by designing a loss function. These methods use a large amount of sample feature information from the head classes and improve the classification performance of the tail classes. In recent years, transfer learning [11–13] has been proposed as a method for long-tailed classification. Here, the goal is to transfer the knowledge learned from the head class to the tail class; large sample feature information from the head class to assist the tail class classification. However, the influence of knowledge transfer is limited, and sometimes knowledge is transferred that misleads tail classification when the features of the head class and the features of the tail class are very different. Recently, data-level strategies, including over-sampling [14–17] and under-sampling [18–21] strategies, have been widely implemented to solve the long-tailed problems. These methods balance the number of samples in the head and tail classes by adjusting

✉ Hong Zhao
hongzhaocn@163.com

[1] School of Computer Science, Minnan Normal University, Zhangzhou 363000, Fujian, China

[2] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou 363000, Fujian, China

Published online: 04 May 2023

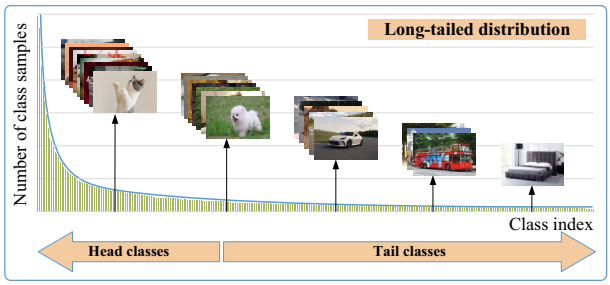


Fig. 1 Structural representation of long-tailed data

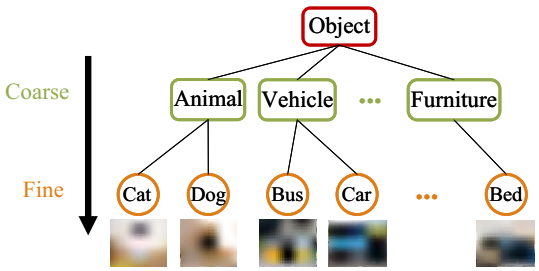


Fig. 2 An example of the hierarchical structure

the samples in different classes. However, under-sampling discards many head class samples, thus losing much feature information. On top of that, over-sampling can lead to the generation of a lot of tail class samples, which includes an overfitting problem.

The above methods are Single Branch Sampling (SBS) methods that presume no semantic relationship between classes [22]. Semantic relations are constructed from the word meaning relations in WordNet, which are widely used in many applications and serve as valuable information to assist in long-tailed classification [23, 24]. Figure 2 gives an example of a hierarchical class structure with five fine-grained and three coarse- classes: the classes *Cat* and *Dog* are fine classes that also belong to the coarse-grained class *Animal*. Similarly, the classes *Bus* and *Car* are fine-grained classes and also belong to the coarse-grained class *Vehicle*. The features required to distinguish between coarse-grained classes differ from those needed for fine-grained classes [25].

In this paper, we present a Coarse-to-Fine knowledge transfer based Bilateral-Sampling Network (CFBSNet) for long-tailed classification. This method mitigates the impact of long-tailed data on the model. The CFBSNet consists of two main parts: network construction and knowledge transfer. First, we propose a bilateral branch sampling network (BBSN) divided into reverse and uniform sampling branches. The two branches obtain samples through different sampling methods and then extract the features of the samples through the unified network. The BBSN strengthens the sampling of the tail class through reverse sampling and concentrates on the tail class. We exploit the hierarchical semantic structure between classes, and the classifiers of two branches are constructed to obtain the coarse- and fine-grained classification probability. Second, we design a coarse-fine knowledge transfer (CFKT) strategy with the ability to tune the different learning stages by adjusting adaptive parameters. The adaptive parameter adjusts according to changes in the training batch, and the weight of coarse- and fine-grained loss is adjusted continuously. Unlike traditional long-tailed classification learning, the CFKT strategy embeds knowledge of coarse-grained classes into models and exploits coarse-grained knowledge to assist with fine-grained classification.

The experiments are conducted on four datasets. We compare the proposed model with several other methods to verify its feasibility. The classification accuracy of the proposed model is 1.89% better than the next-best model. It is 3.16% better than the baseline network on the long-tailed CIFAR-100-LT with an imbalanced ratio of 50. The classification accuracy of the proposed model is 2.13% better than that of the second-best model on the SUN-324 dataset. The Python code written to obtain the experimental results has been uploaded to GitHub (https://github.com/fhqxa/CFBSNet).

The main contributions of this paper can be summarized as follows:

(1) Unlike traditional classification methods with long-tailed distribution, we utilize a bilateral branch sampling structure of uniform and reverse sampling guided by hierarchical semantic relationships. Especially, we trade off the relationship between header and tail classes by fully capturing the information of head class samples, thereby better realizing the transfer of coarse-grained to fine-grained knowledge to address the problem of long-tailed distribution.
(2) Unlike bilateral branch long-tailed classification methods, we design a knowledge transfer strategy from coarse to fine granularity. This strategy gradually transitions the training focus from coarse-grained to fine-grained classification tasks by tuning the loss weight of dual sampling branches, alleviating the imbalance of long-tailed data from the knowledge transfer perspective.
(3) We evaluate the CFBSNet method on multiple long-tailed visual recognition datasets. The experimental results show that the CFBSNet method can fully utilize hierarchical semantic relationships to alleviate the impact of long-tailed data imbalance on model classification performance.

The remainder of this paper is organized as follows. In Sect. 2, related research articles are reviewed. Section 3 presented the details of the proposed model. While Sect. 4 introduces the experimental setup and analyses the experimental results. The main conclusions and ideas for future studies are provided in Sect. 5.

2 Related work

We briefly review algorithm-level strategy, transfer learning method, and data-level strategy are briefly reviewed below.

2.1 Algorithm-level strategy

In recent years, algorithm-level strategies have been extensively studied to alleviate imbalances in long-tailed data by adjusting the weights. Research on algorithm-level strategies is mainly divided into loss functions and other methods. Focal loss [7] adds weight to the loss corresponding to the sample according to the difficulty of sample discrimination. LDAM loss [8] considers the label distribution and encourages the model to have optimal trade-offs between the margins of each class. Additionally, Jamal et al. [9] obtained weights through meta-learning methods to solve the long-tailed problem. Huang et al. [10] learned more discriminative feature representations by preserving intra- and inter-class boundaries, resulting in more balanced class boundaries for data neighborhoods.

The above methods improve the attention to tail classes and the classification performance of tail classes by designing an algorithm to adjust the weight of loss. However, the classification of head classes diminishes when weight loss is performed. The method proposed in this paper exploits a hierarchy from coarse to fine to optimize the model. The loss function of the model fuses the coarse- and fine-grained losses and obtains knowledge that is helpful for fine-grained classification from coarse granularity. The developed method not only strengthens the classification ability of the tail class but also weakens the negative impact on the head class.

2.2 Transfer learning

Transfer learning uses the knowledge learned in the head class to assist with learning the tail class in solving the long-tailed problem. According to the difference in transfer knowledge, transfer learning is mainly divided into object transfer, feature transfer, and parameter transfer. The object transfer method transfers according to the object predicted by the network. The teacher network is trained first, and then the student network is trained. For instance, Xiang et al. [11] fused the loss of the teacher model with the loss of the student model to train the classification ability of the student model. On the contrary, the features of the head samples are transferred to the features of the tail samples. For example, Chu et al. [12] utilized the generic class features of the head as additional knowledge and combined them with the tail class features to enhance the feature distribution of the tail classes. The parameter migration method takes the shared parameters between models as the transfer object. Yin et al. [13] transferred the intra-class variance parameters of the head classes to the tail classes, enhancing the classification ability of the model for the tail classes.

The above horizontal transfer methods improve tail class classification performance by migrating varied head knowledge. These methods easily lead to error transfer because the knowledge systems of the head and tail are different. The vertical transfer method transfers the knowledge containing the same knowledge system from coarse to fine granularity. The proposed method improves transfer effectiveness and reduces the probability of transfer error.

2.3 Data-level strategy

Re-sampling is one data-level strategy used to alleviate the imbalance of long-tailed data. It is divided into over-sampling [26] and under-sampling [27] strategies. Over-sampling methods generate tail samples and make the head and tail samples tend to balance. For instance, Chawla et al. [28] generated tail samples by finding the nearest sample point. Following this, Jiang et al. [14] presented a method to determine the synthetic number of each tail sample according to the classification contribution. Unlike the above method of simply generating tail samples to improve the performance of long-tailed classification, Barua et al. [15] proposed an over-sampling method that synthesizes tail samples by clustering and weighting. Similarly, Han et al. [16] over-sampled the tail samples near the boundary line to improve the effectiveness of the composite samples. In addition, Kim et al. [17] presented a tail sample synthesis method that exploited sufficient samples in the head class to increase the samples of the tail class and enhance the learning ability of the classifier to more general tail class features.

A drawback of under-sampling methods is that they discard some samples of the head class. For instance, Tahir et al. [29] investigated an under-sampling method that constructs a composite boundary between head and tail classes. Under-sampling is sensitive to the data distribution of information at the critical point. Therefore, Ng et al. [18] proposed a sensitivity-based method that obtains distribution information to enhance long-tailed classification performance by clustering header samples. Similarly, Deng et al. [19] proposed an integration algorithm based on automatic clustering, which improves the efficiency of long-tailed classification. The above methods use clustering methods to weaken the impact of long-tailed data, but they are affected by noise interference. Kang et al. [20] proposed combining noise filters to reduce the influence before

under-sampling. In addition, Rekha et al. [21] investigated an under-sampling method that minimizes the excessive elimination of header data and reduces the loss of header information by removing overlapping data points. Furthermore, Zhou et al. [30] proposed a bilateral-branch network that decouples long-tailed learning into the classifier and representational learning, which fuses the two branch features by different sampling strategies.

The above methods mitigate the impact of long-tailed data imbalance on image classification by designing sampling strategies. However, these methods assume that all classes are independent, ignoring the hierarchical relationships between classes. We propose a hierarchical knowledge transfer method based on the semantic relationship between coarse and fine granularity and the hierarchical relationship between classes to mitigate the imbalance problem of long-tailed data.

3 CFBSNet model

In this section, we introduce the proposed Coarse-to-Fine knowledge transfer based Bilateral-Sampling Network (CFBSNet) to allow long-tailed classification.

3.1 Framework overview

We design a Bilateral-Branch Sampling Network (BBSN) based on the Coarse-to-Fine Knowledge Transfer (CFKT) strategy. The coarse-grained data have strong relations with fine-grained data in the long-tailed data [31]. Humans adopt strategies of observing and analyzing problems from different levels in their cognition and processing of real-world problems [32]. Therefore, we consider class granularity in the coarse-to-fine knowledge transfer process to simulate human cognitive processes. The framework of CFBSNet is illustrated in Fig. 3.

The process of the CFBSNet model consists of the following two main stages:

(1) We propose the BBSN long-tailed model that performs uniform sampling and reverses sampling on the long-tailed data. First, the two sampling branches exploit the same module to extract features. Then the features are processed through GAP (Global Average Pooling). Finally, the extracted features were mapped using the softmax function to coarse- and fine-grained classes.
(2) We design a CFKT strategy from coarse to fine granularity. The losses of the two branches are calculated and mixed according to the classification results obtained

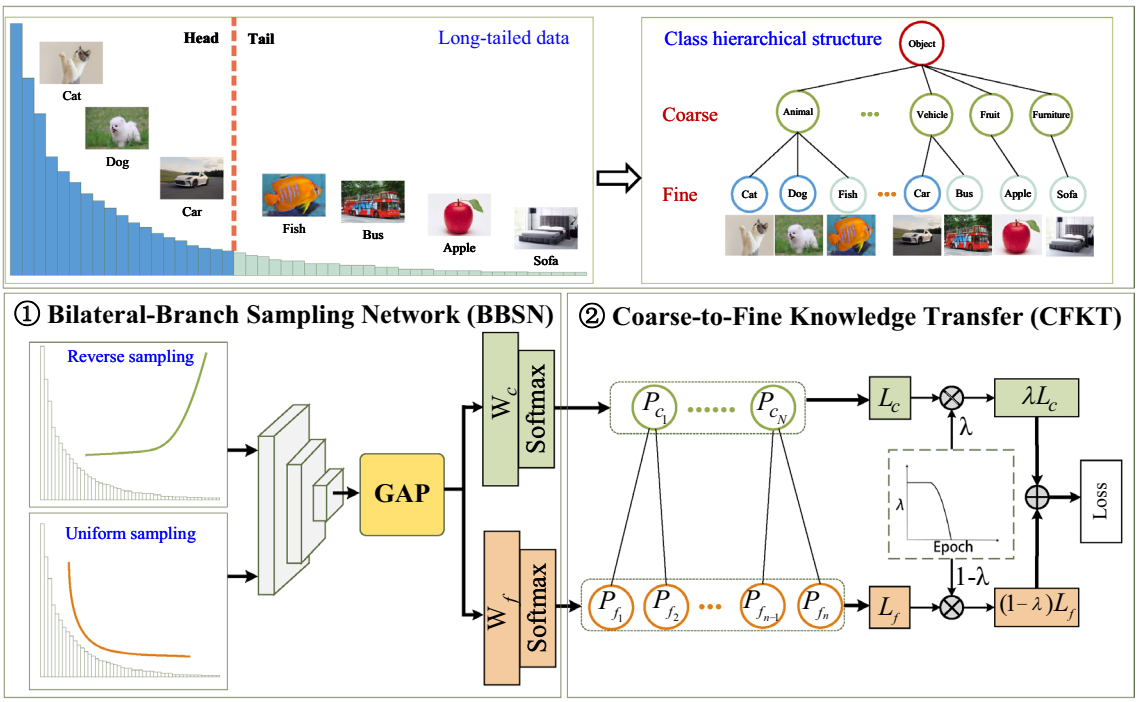


Fig. 3 Framework of the CFBSNet model. The features are processed through GAP (Global Average Pooling). The symbol P_c denotes the coarse-grained classification probability of the fully connected layer. The symbol P_f denotes the coarse-grained classification probability of the fully connected layer

from the full connection layer. The strategy dynamically divides the learning process into three stages: coarse-grained task learning, coarse- and fine-grained mixed learning, and fine-grained task learning. The strategy concentrated adaptively on the different learning according to the training process.

3.2 Bilateral-branch sampling network

The sampling network consists of two branches for sampling long-tailed data: reverse and uniform. The reverse sampling branch aims to alleviate the extreme imbalance, significantly increasing the sampling number of tail samples. The uniform sampling branch obtains each sample with equal probability in a training epoch and retains the characteristics of the original distributions. We fully consider the advantages and characteristics of the two branch sampling methods and appropriately combine them into a bilateral-branch network.

We define total sample set $\mathcal{D} = \{(x_i, y_i, z_i)\}_{i=1}^{N}$ of K classes, where N is the total number of samples, $y \in \{1, \ldots, K\}$ denotes the corresponding fine-grained class label, and $z \in \{1, \ldots, C\}$ denotes the corresponding coarse-grained class label. For two sampling branches, the reverse sampling branch obtains the ith sample (x_i, z_i), and the uniform sampling branch obtains the jth sample (x_j, y_j).

In the uniform sampling branch, each sample is sampled only once with equal probability in the training round. The uniform sampler calculates the sampling probability of class c according to the number of samples. The expression for calculating P_c is expressed as:

$$P_c = \frac{n_c}{\sum_{j=1}^{K} n_j}, \tag{1}$$

where K is the number of classes, n_j is the number of samples of the jth class, and P_c is the probability of sampling of the ith class.

In the reverse sampling branch, the sampling probability of each class is directly proportional to the reciprocal of its sample size. The more samples in a class, the smaller the sampling probability of that class. The sampling probability of class f is calculated according to the number of samples. The expression for calculating P_f is expressed as:

$$P_f = \frac{w_f}{\sum_{j=1}^{K} w_j}, \tag{2}$$

where P_f is the probability of drawing the fth class and w_f is

$$w_f = \frac{N_{max}}{N_f}, \tag{3}$$

where N_{max} is the maximum number of samples of all classes and N_f is the number of samples of the fth class.

The histogram of the number of samples for uniform sampling and reverse sampling is shown in Fig. 4. The abscissa represents the class index, and the ordinate represents the number of samples from the corresponding class. The reverse sampling method emphasizes the tail class because of the imbalance in long-tailed data. The uniform sampling method preserves the characteristics of the original distribution because each sample is sampled with the same probability.

3.3 Coarse-to-fine knowledge transfer strategy

In this section, we investigate a coarse-to-fine knowledge transfer strategy (CFKT) by integrating coarse-grained and fine-grained loss functions into a framework.

According to the CFKT strategy, we can divide the training process into three stages. (1) The first stage is the coarse-grained training stage. The model is pre-trained to learn the parameters for the fine-grained classification of the model. (2) The second stage is coarse and fine-granularity mixed learning. The knowledge learned in the coarse-grained learning stage is constantly generalized and transferred to fine-grained training. Coarse- and fine-grained learning benefit each other after the first two stages of training. (3) The third stage is the fine-grained training stage. It strengthens the training in fine

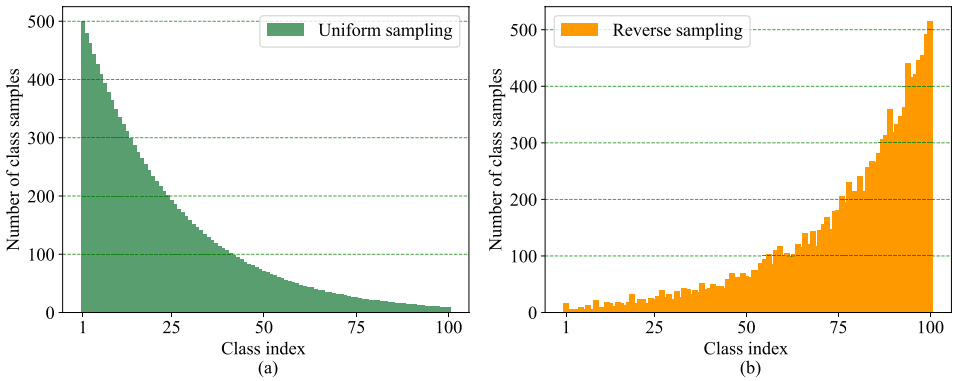


Fig. 4 Sample distribution diagram of reverse and uniform sampling on CIFAR-100-LT

granularity based on the pre-training model and enhances the classification ability of the model.

The reverse sampling branch performs coarse-grained learning, while the uniform sampling branch performs fine-grained learning. Initially, we use the convolutional neural network (CNN) and GAP to extract and process features, respectively. GAP [33] can better correspond the classes to the feature map of the last convolution layer, reducing the number of parameters and integrating the global spatial information. The GAP calculates an average value of all pixels of the feature map of each output channel to obtain the feature vector and input it to the softmax layer. Let f_ϕ be the feature extraction module. We feed samples x_i and x_j into the feature extraction module to obtain the coarse-grained sample feature $f_\phi(x_i)$ and fine-grained sample feature $f_\phi(x_j)$. The softmax function maps the extracted features to coarse- and fine-grained classes.

For each coarse-grained class $i \in \{1, \ldots, C\}$, the softmax function calculates the probability of the class i using the follow equation:

$$\hat{p}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \tag{4}$$

where $z_i = W_c(\phi(x_i))$ is the coarse-grained predicted output. The output probability distribution of the coarse-grained task is denoted as $\hat{\mathbf{p}}_c = [\hat{p_1}, \hat{p}_2, \ldots, \hat{p_C}]^{\mathrm{T}}$, where $\hat{p_C}$ represents the prediction probability of each coarse-grained class.

For each fine-grained class $s \in \{1, \ldots, K\}$, the softmax function calculates the probability of class s as follow:

$$\hat{p}_s = \frac{e^{z_s}}{\sum_{j=1}^{K} e^{z_j}}, \tag{5}$$

where $z_s = W_f(\phi(x_j))$ is the coarse-grained predicted output. The output probability distribution of the coarse-grained task is denoted as $\hat{\mathbf{p}}_f = [\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_K]^{\mathrm{T}}$, where $\hat{p}_K$ represents the prediction probability of each fine-grained class.

In the CFKT strategy, the total loss is obtained by the weighted addition of the losses of the two branches. We split the training process into three stages by controlling the weight of the loss caused by the two branches: coarse-grained task learning, coarse- and fine-grained learning, and fine-grained task learning. Researchers observe, analyze, and solve problems from different levels and perspectives when encountering real-world problems [32, 34]. In most cases, humans can use the concepts of hierarchy and granularity to perceive and understand the world [35]. The category granularity relationship from simple to deep is considered to simulate the human cognitive process of a linear combination of loss of coarse- and fine-grained classes. The method transfers knowledge from coarse-grained to fine-grained. The weighted cross-entropy classification loss of CFBSNet is as follows:

$$\mathcal{L} = \lambda E(\hat{\mathbf{p}}_c, y_c) + (1 - \lambda) E(\hat{\mathbf{p}}_f, y_f), \tag{6}$$

where λ is the weight that accounts for the loss caused by the two branches, $E(\hat{\mathbf{p}}_c, y_c)$ represents the loss of coarse-grained learning, $E(\hat{\mathbf{p}}_f, y_f)$ represents the loss of fine-grained learning. These two losses are used to learn coarse- and fine-grained tasks. In the training process, the training attention will be gradually transferred from coarse- to fine-grained learning with the change of the parameter λ. Adjusting the weight of coarse- and fine-grained loss is dynamically mixed into the optimization target loss $\mathcal{L}$.

In this respect, the CFKT strategy controls and switches learning stages by adjusting the weight of coarse- and fine-grained losses. Kendall et al. [36] thought that loss weight is a very important factor in the learning process. Task parameters with larger loss weights update faster than those with smaller loss weights. Therefore, the CFKT passes the weight parameter λ to adjust the contribution of coarse- and fine-grained loss to parameter updates. In the first stage, only learning coarse-grained tasks are considered when the parameter λ was set to 1. At this time, the loss weight of fine-grained tasks is 0, representing that no parameter update has been performed. In the second stage, parameter λ gradually reduces to 0 with the increase of the training batch, which represents learning two tasks simultaneously and transferring coarse-grained knowledge to fine-grained knowledge. The final stage is contrary to the point of attention in the first stage. The parameter λ is set to 0, which does not update the parameters of the coarse-grained tasks and adjust the fine-grained task. Parameter λ can be expressed as follows:

$$\lambda = \begin{cases} 1, & 0 < T \le T_c; \\ 1 - \left(\frac{T - T_c}{T_{max} - T_c - T_f - 1}\right)^2, & T_c < T \le T_{max} - T_f; \\ 0, & T_{max} - T_f < T \le T_{max}, \end{cases} \tag{7}$$

where T_c and T_f are the training epoch numbers in coarse- and fine-grained training, respectively. Parameter T refers to

Table 1 Descriptions of the experimental datasets

Dataset	Coarse-grained class	Fine-grained class	Training size	Imbalance way
CIFAR-100-IR10	20	100	19,572	Artificial
CIFAR-100-IR50	20	100	12,607	Artificial
CIFAR-100-IR100	20	100	10,847	Artificial
VOC	3	19	2937	Real-world
SUN	15	324	85,147	Real-world
iNaturalist	13	5089	579,184	Real-world

the number of the current epoch. Parameter T_{max} is the maximum number of epochs. The reason for design parameter λ is that the model immediately turns its attention to fine-grained tasks when coarse-grained tasks are well-trained, thus becoming a classic model without hierarchy.

Algorithm 1 provides the pseudo-code for the model training process. Sets of bilateral samples were constructed in lines 4–6 and extracted features in line 7. The probabilities of the coarse- and fine-grained samples were calculated in lines 8 and 10, and the losses were calculated in lines 9 and 11. After that, the loss was fused in 12 followed by the parameters ϕ, $\mathbf{W}_c$ and $\mathbf{W}_f$ being updated by loss $\mathcal{L}$ backpropagation in line 13.

Algorithm 1 Coarse-to-Fine knowledge transfer based Bilateral-Sampling Network (CFBSNet) for long-tailed classification

Input: Training set $\mathcal{D} = (x_i, y_i, z_i)_{i=1}^{N}$ of K classes, where N is the total number of samples, $y \in \{1,...,K\}$ denotes the corresponding fine-grained class label, and $z \in \{1,...,C\}$ denotes the corresponding coarse-grained class label. $\mathbf{X}$ is the sample set. The number of training epochs is E_{max}. The batch of each epoch is B_{max}. $\mathbf{X}_c$ and $\mathbf{X}_f$ are sample sets obtained by corresponding sampling strategy. The feature extraction module is f_ϕ. Matrixes $\mathbf{W}_c$ and $\mathbf{W}_f$ are the parameters of coarse- and fine-grained fully connected layers in decibels.

Output: The parameters ϕ, $\mathbf{W}_c$ and $\mathbf{W}_f$.

1: Initialize parameters ϕ, $\mathbf{W}_c$ and $\mathbf{W}_f$;
2: **for** $epoch = 1 : E_{max}$ **do**
3: **for** $batch = 1 : B_{max}$ **do**
4: Calculate reverse sampling branch sampling probability P_c according to Eq. (1);
5: Calculate uniform sampling branch sampling probability P_f according to Eq. (2);
6: Obtain new sample sets $\mathbf{X}_c$ and $\mathbf{X}_f$ by sampling $\mathbf{X}$;
7: Extract the $\mathbf{X}_c$ and $\mathbf{X}_f$ features by f_ϕ respectively;
8: Calculate probability of coarse-grained class $\hat{\mathbf{p}}_c$ according to Eq. (4);
9: Obtain loss of coarse-grained class $\mathcal{L}_c$;
10: Calculate probability of fine-grained class $\hat{\mathbf{p}}_f$ according to Eq. (5);
11: Obtain loss of fine-grained class $\mathcal{L}_f$;
12: Calculate the loss $\mathcal{L}$ according to Eq. (6);
13: Update parameters ϕ, $\mathbf{W}_c$ and $\mathbf{W}_f$ by loss $\mathcal{L}$ backpropagation;
14: **end for**
15: **end for**

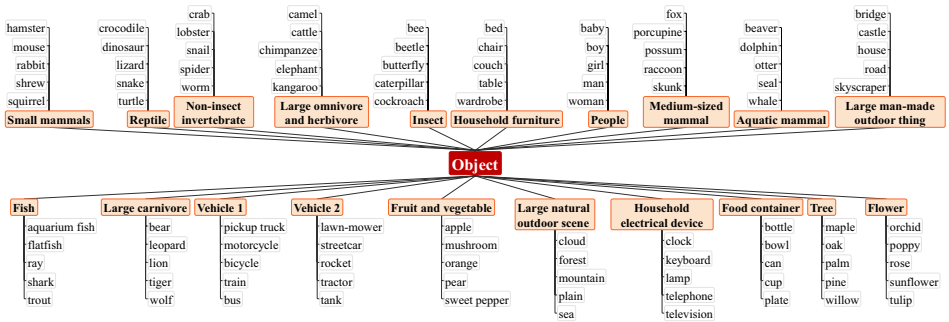
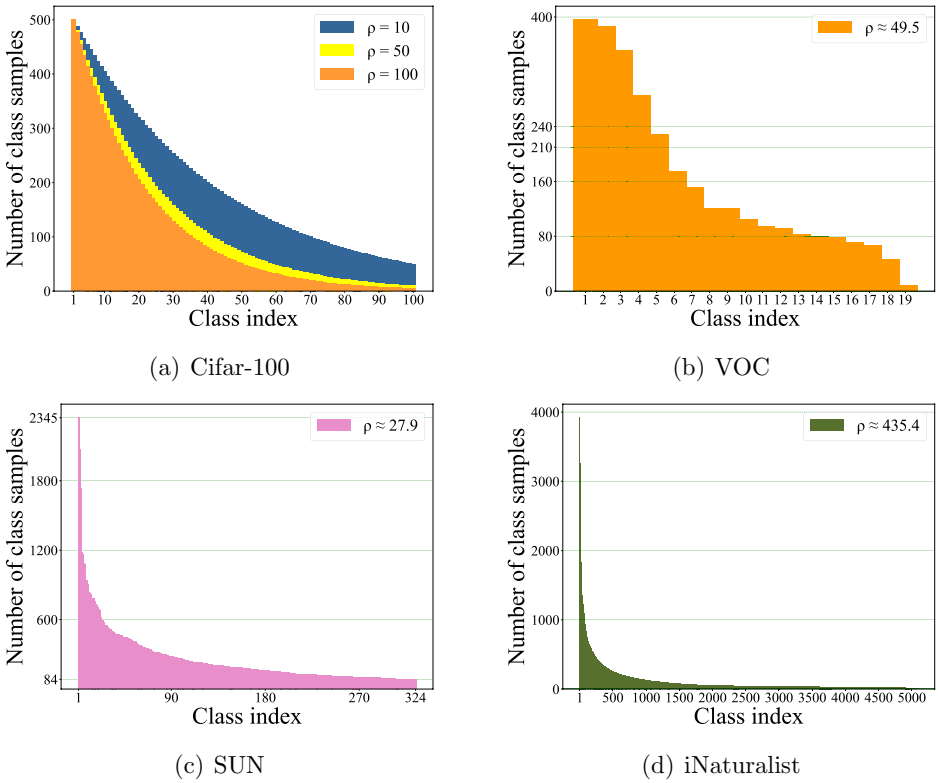


Fig. 5 The class hierarchy semantic structure of 20 coarse-grained classes and 100 fine-grained classes on the CIFAR-100 dataset



(a) Cifar-100 (b) VOC

(c) SUN (d) iNaturalist

Fig. 6 Histograms of simple number of each class on the CIFAR-100, SUN, VOC, and iNaturalist datasets

The method uses a general convolutional neural network framework to extract features, pool by global average, and perform operations through the full connection layer. The method designs a specific loss function to adjust the training of the framework through weight distribution. This framework is applied to the existing methods such as Focal [7] and LDAM [8] losses. For example, Focal loss decreases the losses of head classes, which optimizes the training process of the framework. We design a training framework for parameter sharing, where two branches are trained simultaneously, and the backbone network of the two branches shares parameters. This parameter-sharing method is also applied to existing long-tailed classification methods. For example, the Self Supervision to Distillation [37] method enhances the effect of knowledge transfer through parameter sharing to reduce the impact of long-tailed data on the model. In addition, we perform the same experimental setup as existing methods. The proposed method is at the same level as the existing methods for time efficiency.

4 Experimental settings and results analysis

In this section, (1) we introduce the experimental datasets; (2) we introduce the compared methods; (3) we analyze the effectiveness of parameter λ in the training stage; (4)

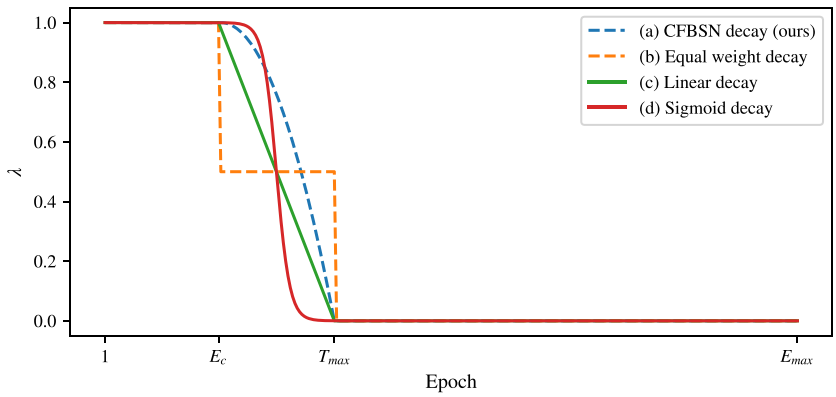


Fig. 7 Function curves of different functions

Table 2 The ACC (%) comparison of the parameter λ on CIFAR-100-LT

Dataset	Long-tired CIFAR-100	
Imbalance ratio	$\rho = 50$	$\rho = 100$
Equal weight	43.41	38.45
Linear decay	43.56	38.29
Sigmoid decay	43.12	37.89
CFBSNet decay (ours)	**44.69**	**39.09**

The best results are marked in bold

we explore the performance of the CFKT strategy; (5) we perform the efficiency of transfer from reverse sampling to uniform sampling; (6) we compare the performance of CFBSNet in local accuracy; (7) we compare CFBSNet with other methods; (8) we explore the statistical test of different methods.

Resnet32 [38] was used as the backbone in all the comparative experiments. The specific experiment settings were as follows: all experiments use a standard small-batch stochastic gradient descent (i.e., SGD) [39] with a weight decay of 2×10^{-4} and a momentum of 0.9. The number of training epochs is 300, and the batch size is 128. We train the first five epochs using the linear warm-up learning rate plan. The initial learning rate is set to 0.1, and the learning rate is attenuated by 0.01 at 180 and 230 epochs, respectively. To better check the performance changes of a different course number during training, we classify the classes based on the number of images. These classes are divided into three groups: the head classes (over 100 images), the middle classes (20–100 images), and the tail classes (1–20 images). We train all models on a Windows 10 desktop computer using NVIDIA GeForce RTX 2080 Ti with 11.0 GB video memory and a 3.60 GHz CPU.

4.1 Datasets

The experiments use four long-tailed datasets, including CIFAR-100-LT, SUN, VOC, and iNaturalist. Table 1 provides descriptions for the four datasets. This structure is constructed by the semantic dependency between classes in WordNet [40–42]. Figure 5 shows the class hierarchy semantic structure of 20 coarse-grained and 100 fine-grained classes on the CIFAR-100 dataset.

CIFAR-100-LT. The CIFAR-100 [43] dataset consists of 60,000 color images and is divided into 100 classes. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. CIFAR-100-LT is processed by CIFAR-100 and contains the same classes. The same version of CIFAR-100 was used for the fairness of experimentation. The imbalance degree of the data is determined by the imbalance coefficient α control.

Table 3 The ACC (%) comparison between CFKT and flat methods

Dataset		Long-tired CIFAR-100		
Imbalance ratio		$\rho = 10$	$\rho = 50$	$\rho = 100$
CE	Flat	56.45	42.96	38.12
CE	CFKT	**56.88**	**43.77**	**38.46**

The best results are marked in bold

Table 4 The ACC (%) comparison between reverse transfer to uniform transfer and uniform methods

Dataset		Long-tired CIFAR-100		
Imbalance ratio		$\rho = 10$	$\rho = 50$	$\rho = 100$
CFBSNet	Uniform	**56.29**	42.25	38.29
CFBSNet	Reverse transfer to uniform	56.12	**44.69**	**39.09**

The best results are marked in bold

Unbalance coefficient α is defined as N_{max}/N_{min} on the dataset, the imbalance rates of the experiment α were 10, 50 and 100. Figure 6a illustrates distributions of classes for the CIFAR-100 dataset on three imbalance rates.

VOC. The 2012 version of VOC [44] contains 20 classes. Some of these classes are multi-label classes; therefore, we remove these multi-label classes from this dataset. Figure 6b illustrates distributions of classes for VOC.

SUN. The extensive Scene UNderstanding (SUN) [45] database contains 397 classes. We deleted the multi-label classes on SUN-397 and retained 324 classes. Figure 6c illustrates distributions of classes for SUN. All fine-grained classes are grouped into 15 coarse-grained classes.

iNaturalist. The 2017 version of iNaturalist [46] contains 675,170 images with 5089 classes built by a three-level knowledge graph. We use the official split containing 579,184 training and 95,986 test images. Figure 6d illustrates distributions of classes for iNaturalist.

4.2 Comparison methods

We compare CFBSNet with several models, the details of which are as follows:

Loss functions:

(1) Cross-entropy loss (CE): The weight of each sample is the same, and the loss of CE is used for standard training.
(2) Focal loss (Focal) [7]: Focal is modified based on the standard cross-entropy loss. The model focuses more on tail samples during training by reducing the weight of head samples.
(3) Label-distribution-aware margin loss (LDAM) [8]: LDAM loss is considered from the perspective of regularization and encourages larger margins for the tail class.

Strategies:

(1) Re-weighting (RW) [47]: The reciprocal number of samples in each class is used as the weight to balance the loss of long-tailed samples.
(2) Deferred re-weighting (DRW) [8]: The DRW strategy defers the re-weighting strategy to a later stage of training and is usually combined with a loss function to form an algorithm.

4.3 Effectiveness of parameter λ in training stage

In this section, we verify the effectiveness of CFBSNet decay as parameter λ. Parameter λ is indispensable in the coarse and fine mixed training stage, which affects the transition from coarse- to fine-grained. We compare CFBSNet decay with three decay functions as parameter λ. The three decay functions are equal weight decay, linear decay, and sigmoid decay. The function curve of parameter λ is varying with batch as shown in Fig. 7.

As can be observed from Fig. 7, the coarse-grained weight loss was visualized as a function curve. The curves of the different decay functions were plotted. The equal weight decay function yielded the same weight as the thickness loss in the thickness mixed learning stage. The linear decay function transitions from coarse- to fine-grained learning at a uniform decay rate. The sigmoid decay function decays rapidly in a very short batch interval. The CFKT strategy changes with the batch when CFBSNet decays weight transfer. The weight transfer between coarse- and fine-grained losses is more stable than equal weight decay and sigmoid decay. The proposed method focuses more on coarse-grained learning than the linear decay strategy to assist fine-grained learning.

The experimental results of different parameter values λ affecting the classification accuracy are shown in

Table 5 Comparison of CIFAR-100-LT classification local ACC (%)

Dataset	Long-tired CIFAR-100			
Imbalance ratio	$\rho = 50$			
Method	Head(41)	Middle(40)	Tail(19)	All(100)
Focal	65.17	35.43	12.42	43.25
LDAM	**65.68**	35.93	12.53	43.68
LDAM-DRW	65.46	36.78	14.11	44.23
CFBSNet(ours)	63.44	**40.75**	**20.05**	**46.12**

The best results are marked in bold

Table 6 The ACC (%) of CIFAR-100-LT classification performance under different unbalance ratios

Dataset	Long-tired CIFAR-100		
Imbalance ratio	$\rho = 10$	$\rho = 50$	$\rho = 100$
CE [48]	56.45	42.96	38.12
Focal [7]	56.09	43.25	38.39
LDAM [8]	56.5	43.68	39.43
LDAM-DRW [8]	**57.18**	44.23	39.81
MCKT [49]	57.15	44.24	39.15
CFBSNet(ours)	56.87	**46.12**	**40.64**

The best results are marked in bold

Table 7 The ACC (%) of SUN, VOC, and iNaturalist classification performance under different unbalance ratios

Dataset	SUN	VOC	iNaturalist
Imbalance ratio	$\rho \approx 27.9$	$\rho \approx 49.5$	$\rho \approx 435.4$
CE [48]	36.84	**40.78**	7.08
Focal [7]	37.33	40.65	7.26
LDAM [8]	32.23	37.92	**7.96**
LDAM-DRW [8]	34.80	38.09	7.26
MCKT [49]	35.49	38.89	7.11
CFBSNet(ours)	**39.46**	39.34	7.32

The best results are marked in bold

Table 2. The results are consistent with the analysis of the function curves in Fig. 7. Compared with the other three decay functions, the CFBSNet decay function has improved performance on the CIFAR-100-LT dataset. For instance, it is 1.13% better than the second place when the imbalance rate is 50. CFBSNet decay function helps to transfer from coarse to fine step by step in the coarse- and fine-grained mixed training stage.

4.4 Efficiency of CFKT strategy

We compare the performance of the flat strategy and the CFKT strategy. For the flat strategy, the loss calculated according to the classification prediction results is the final loss. For the CFKT strategy, we use semantic information to build a hierarchical structure. The network predicts and classifies the extracted features into coarse- and fine-grained classes, respectively, and calculates the corresponding loss. Then we use the parameters described in the previous section λ and add coarse and fine granularity loss weights. Finally, the learning of an epoch was completed through gradient feedback. The applied strategy only was changed, and the other settings remained unchanged to determine the fairness of the experiment. The experimental results are shown in Table 3.

On the CIFAR-100-LT dataset, the classification accuracy of the CFKT strategy is better than that of the flat strategy by 0.43%, 0.91%, and 0.34% when the imbalance rate is 10, 50, and 100, respectively. The CFKT strategy fully considers the interrelated characteristics of data and uses the semantic relationships between the data to improve the classification performance of the model.

4.5 Efficiency of transfer from reverse to uniform sampling

In this section, the effectiveness of the reverse sampling strategy in improving the performance of the model was analyzed. Reverse sampling pays more attention to the tail samples, and uniform sampling follows the law of sampling in real life. We use the sampling method of coarse-grained branches as variables while other experimental settings remain the same. Transferring reverse sampling to uniform sampling is compared with the traditional uniform sampling method. The experimental results are shown in Table 4.

The following analysis can be obtained: transferring from reverse sampling to uniform sampling achieves good results compared with the traditional uniform sampling method. Although the accuracy is lower than that of the uniform sampling method when the unbalance rate is 10, the accuracy is improved by 2.44% and 0.8% when the unbalance rate is 50 and 100, respectively. The sampling strategy shows a good effect in the experiment. Reverse sampling was also used to learn the coarse-grained knowledge to assist with the fine-grained classification.

Table 8 Average ranks of ACC of different methods on six datasets

	CE	Focal	LDAM	LDAM-DRW	MCKT	CFBSNet
CIFAR-IR10	56.45(5)	56.09(6)	56.50(4)	**57.18(1)**	57.15(2)	56.87(3)
CIFAR-IR50	42.96(6)	43.25(5)	43.68(4)	44.23(3)	44.24(2)	**46.12(1)**
CIFAR-IR100	38.12(6)	38.39(5)	39.43(3)	39.81(2)	39.15(4)	**40.64(1)**
SUN	36.84(3)	37.33(2)	32.23(6)	34.80(5)	35.49(4)	**39.46(1)**
VOC	40.78(1)	40.65(2)	37.92(6)	38.09(5)	38.89(4)	39.34(3)
iNaturalist	7.08(6)	7.26(3)	**7.96(1)**	7.26(3)	7.11(5)	7.32(2)
Avg. rank	4.50	3.83	4.00	3.17	3.50	1.83

4.6 Efficiency of CFBSNet in local accuracy

In this section, we explore the classification accuracy of local classes on the CIFAR-100-LT dataset to illustrate the validity of CFBSNet. The CIFAR-100-LT dataset with an imbalance rate of 50 is divided into three groups according to the experimental setting: head (41 classes), middle (40 classes), and tail (19 classes). We compared the CFBSNet method with three methods: Focal, LDAM, and LDAM-DRW. The experimental results are shown in Table 5.

As seen from Table 5, CFBSNet improved the overall classification accuracy (all), which is 1.43% and 1.89% better than the second and third place, respectively. From the perspective of local accuracy, the CFBSNet classification accuracy of the head, middle, and tail classes was 63.44%, 40.75%, and 20.05%, respectively. CFBSNet is 3.97% better than the second place in the middle and 4.89% better than the second place in the tail. The classification accuracy of CFBSNet is 2% lower than that of the first place in the head.

From the above results, it can be argued that CFBSNet sacrifices the classification accuracy of some heads but enhances middle and tail classification performance. The CFBSNet method improves the overall classification accuracy and can effectively alleviate the impact of data imbalance in the long-tailed data environment.

4.7 Efficiency of CFBSNet comparison with other methods

In this section, we compare CFBSNet with five methods: CE, Focal, LDAM, CFKT, and LDAM-DRW. We conduct experiments on the VOC, CIFAR-100-LT, iNaturalist, and SUN datasets to prove the applicability and feasibility of the proposed model. The experimental results are listed in Tables 6 and 7.

As can be seen from the experimental results:

(1) On CIFAR-100, the CFBSNet is almost superior to the other five methods when the unbalance rates are 10, 50, and 100. The effect is best when the unbalance rate is 50. The prediction accuracy is 46.12%, 1.89% better than that of second place. The experimental results show that the CFBSNet method is effective in promoting classification.

(2) The prediction accuracy of the CE method is 40.78% and achieves the highest prediction accuracy on VOC. The prediction accuracy of CFBSNet is 1.44% lower than CE and ranks third among all comparison methods. The class structure only partially reflects the relationship between coarse-grained and fine-grained because it has 19 fine-grained and three coarse-grained classes. An error can be easily caused when knowledge is transferred between granularities; therefore, CFBSNet could achieve better classification performance on VOC.

(3) CFBSNet shows good results on the SUN dataset and ranked first among all comparison methods. The prediction accuracy of CFBSNet is 39.46%, which is better than the second at 2.13%. The SUN dataset has 324 classes with rich semantic structures among them. The proposed method can effectively improve classification ability by using the relationship between coarse and fine granularity.

(4) CFBSNet has a prediction accuracy of 7.32% and ranks second in six methods on iNaturalist. The prediction accuracy of LDAM is 0.64% better than CFBSNet and achieves the highest prediction probability. The iNaturalist dataset has 5089 fine-grained classes, but the number of fine-grained classes is only 34. The huge difference in the number of categories easily leads to misleading knowledge. Therefore, CFBSNet does not achieve the best classification performance on iNaturalist.

4.8 Statistical test of different methods

We perform statistical tests on CFBSNet and other comparison methods. A statistical test is used to analyze the performance of the proposed method. We conduct experiments on K comparison methods and N experimental datasets. We calculate the average ranking according to the accuracy (ACC) for different comparison methods. We calculate the average ranking of each method on all datasets as follows:

$$R_i = \frac{1}{N}\sum_{j=1}^{N} r_j^i, \tag{8}$$

where r_j^i represents the ranking of the ith method on the jth dataset. The average ranking of ACC of K methods is shown in Table 8.

From the results in Table 8, we can obtain that CFBSNet is superior to all comparison methods in average accuracy. The average ranking of CFBSNet is 1.34 higher than that in second place and 2.67 higher than that in last place. Therefore, the experimental and statistical results show that CFBSNet performed highly competitively compared with the other methods.

5 Conclusions and future work

In this paper, we proposed a Coarse-to-Fine knowledge transfer based Bilateral-Sampling Network (CFBSNet) for long-tailed classification. The developed techniques can significantly alleviate the impact of long-tailed data imbalances on classification. CFBSNet can consider the coarse- and fine-grained semantic relationship between data, which is helpful for the long-tailed classification task. Unlike the traditional single-branch long-tailed classification method, CFBSNet proposes a Bilateral-Branch Sampling Network (BBSN) and Coarse-to-Fine Knowledge Transfer (CFKT) strategy. The BBSN comprises bilateral sampling branches, and different sampling strategies are used to reduce the impact of insufficient tail data. The CFKT strategy lets coarse-grained learning assist fine-grained classification and improves the classification ability of the model. The experimental results show that the CFBSNet model is comparable to several advanced long-tailed models. The semantic structure with hierarchical relationships played an indispensable role in the training. In the future, we will focus on transferring knowledge from the head to the tail and building hierarchical relationships through clustering.

Acknowledgements This work was supported by the Natural Science Foundation of Fujian Province under Grant no. 2021J011003 and the National Natural Science Foundation of China under Grant no. 62141602.

Data availability Some or all data, models, or code generated or used during the study are available from the Data availability corresponding author by request (Hong Zhao).

Declarations

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

References

1. Cao Y, Kuang J, Gao M, Zhou A, Wen Y, Chua T (2023) Learning relation prototype from unlabeled texts for long-tail relation extraction. IEEE Trans Knowl Data Eng 35(2):1761–1774
2. Wu T, Liu Z, Huang Q, Wang Y, Lin D (2021) Adversarial robustness under long-tailed distribution. In: IEEE/CVF conference on computer vision and pattern recognition, pp 8659–8668
3. Goecks J, Jalili V, Heiser LM, Gray JW (2020) How machine learning will transform biomedicine. Cell 181(1):92–101
4. Kulkarni R, Di Minin E (2021) Automated retrieval of information on threatened species from online sources using machine learning. Methods Ecol Evol 12(7):1226–1239
5. Zeng D, Veldhuis R, Spreeuwers L (2021) A survey of face recognition techniques under occlusion. IET Biom 10(6):581–606
6. Haggag M, Siam AS, El-Dakhakhni W, Coulibaly P, Hassini E (2021) A deep learning model for predicting climate-induced disasters. Nat Hazards 107(1):1009–1034
7. Lin T, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: IEEE international conference on computer vision, pp 2980–2988
8. Cao K, Wei C, Gaidon A, Arechiga N, Ma T (2019) Learning imbalanced datasets with label-distribution-aware margin loss. Adv Neural Inf Process Syst 32:1–8
9. Jamal MA, Brown M, Yang M, Wang L, Gong B (2020) Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In: IEEE/CVF conference on computer vision and pattern recognition, pp 7610–7619
10. Huang C, Li Y, Loy CC, Tang X (2019) Deep imbalanced learning for face recognition and attribute prediction. IEEE Trans Pattern Anal Mach Intell 42(11):2781–2794
11. Xiang L, Ding G, Han J (2020) Learning from multiple experts: self-paced knowledge distillation for long-tailed classification. In: European conference on computer vision, pp 247–263
12. Chu P, Bian X, Liu S, Ling H (2020) Feature space augmentation for long-tailed data. In: European conference on computer vision, pp 694–710
13. Yin X, Yu X, Sohn K, Liu X, Chandraker M (2019) Feature transfer learning for face recognition with under-represented data. In: IEEE/CVF conference on computer vision and pattern recognition, pp 5704–5713
14. Jiang Z, Pan T, Zhang C, Yang J (2021) A new oversampling method based on the classification contribution degree. Symmetry 13(2):194
15. Barua S, Islam MM, Yao X, Murase K (2012) MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans Knowl Data Eng 26(2):405–425
16. Han H, Wang W, Mao B (2005) Borderline-smote: a new oversampling method in imbalanced data sets learning. In: International conference on intelligent computing, pp 878–887
17. Kim J, Jeong J, Shin J (2020) M2m: imbalanced classification via major-to-minor translation. In: IEEE/CVF conference on computer vision and pattern recognition, pp 13896–13905
18. Ng WW, Hu J, Yeung DS, Yin S, Roli F (2014) Diversified sensitivity-based undersampling for imbalance classification problems. IEEE Trans Cybern 45(11):2402–2412
19. Deng X, Zhong W, Ren J, Zeng D, Zhang H (2016) An imbalanced data classification method based on automatic clustering under-sampling. In: IEEE International performance computing and communications conference, pp 1–8
20. Kang Q, Chen X, Li S (2016) A noise-filtered under-sampling scheme for imbalanced classification. IEEE Trans Cybern 47(12):4263–4274
21. Rekha G, Reddy VK, Tyagi AK (2020) Critical instances removal based under-sampling (CIRUS): a solution for class imbalance problem. Int J Hybrid Intell Syst 16(2):55–66
22. Xu H, Zhang X, Li H, Xie L, Dai W, Xiong H, Tian Q (2022) Seed the views: hierarchical semantic alignment for contrastive representation learning. IEEE Trans Pattern Anal Mach Intell 45(3):3753–3767
23. Li S, Gong K, Liu CH, Wang Y, Qiao F, Cheng X (2021) Metasaug: meta semantic augmentation for long-tailed visual recognition. In: IEEE/CVF conference on computer vision and pattern recognition, pp 5212–5221
24. Xu W, Yuan K, Li W, Ding W (2023) An emerging fuzzy feature selection method using composite entropy-based uncertainty measure and data distribution. IEEE Trans Emerg Top Comput Intell 7(1):76–88
25. Li J, Li Y, Mi Y, Wu W (2020) Meso-granularity labeled method for multi-granularity formal concept analysis. J Comput Res Dev 57(2):447–458

26. Liu R (2022) A novel synthetic minority oversampling technique based on relative and absolute densities for imbalanced classification. Appl Intell 53(1):768–803
27. Buda M, Maki A, Mazurowski MA (2018) A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw 106:249–259
28. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
29. Tahir MA, Kittler J, Yan F (2012) Inverse random under sampling for class imbalance problem and its application to multi-label classification. Pattern Recognit 45(10):3738–3750
30. Zhou B, Cui Q, Wei X, Chen Z (2020) Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: IEEE/CVF conference on computer vision and pattern recognition, pp 9719–9728
31. Yao Y (2004) A partition model of granular computing. LNCS Trans Rough Sets I, LNCS 3100:232–253
32. Yao Y (2008) Granular computing: past, present and future. In: IEEE international conference on granular computing, pp 80–85
33. Chen Q, Liu Q, Lin E (2021) A knowledge-guide hierarchical learning method for long-tailed image classification. Neurocomputing 459:408–418
34. Xu W, Guo D, Qian Y, Ding W (2022) Two-way concept-cognitive learning method: a fuzzy-based progressive learning. IEEE Trans Fuzzy Syst 1–15
35. Xu W, Pan Y, Chen X, Ding W, Qian Y (2022) A novel dynamic fusion approach using information entropy for interval-valued ordered datasets. IEEE Trans Big Data 1–15
36. Kendall A, Gal Y, Cipolla R (2018) Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: IEEE conference on computer vision and pattern recognition, pp 7482–7491
37. Li T, Wang L, Wu G (2021) Self supervision to distillation for long-tailed visual recognition. In: IEEE/CVF international conference on computer vision, pp 630–639
38. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: European conference on computer vision, pp 630–645
39. Bottou L (2012) Stochastic gradient descent tricks. In: Montavon G, Orr GB, Müller KR (eds) Neural networks: tricks of the trade. Springer, Berlin, pp 421–436
40. Xu W, Guo D, Mi J, Qian Y, Zheng K, Ding W (2023) Two-way concept-cognitive learning via concept movement viewpoint. IEEE Trans Neural Netw Learn Syst 1–15
41. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
42. Yuan K, Xu W, Li W, Ding W (2022) An incremental learning mechanism for object classification based on progressive fuzzy three-way concept. Inf Sci 584:127–147
43. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778
44. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The Pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
45. Zhou B, Lapedriza A, Torralba A, Oliva A (2017) Places: an image database for deep scene understanding. J Vis 17(10):296–296
46. Cui Y, Song Y, Sun C, Howard A, Belongie S (2018) Large scale fine-grained categorization and domain-specific transfer learning. In: IEEE conference on computer vision and pattern recognition, pp 4109–4118
47. Huang C, Li Y, Loy CC, Tang X (2016) Learning deep representation for imbalanced classification. In: IEEE conference on computer vision and pattern recognition, pp 5375–5384
48. De Boer P, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. Ann Oper Res 134(1):19–67
49. Li Z, Zhao H, Lin Y (2022) Multi-task convolutional neural network with coarse-to-fine knowledge transfer for long-tailed classification. Inf Sci 608:900–916