Neural Processing Letters (2024) 56:158
https://doi.org/10.1007/s11063-024-11624-x

Class-Balanced Regularization for Long-Tailed Recognition

Yuge Xu[1] · **Chuanlong Lyu**[1]

Accepted: 10 April 2024

Abstract
Long-tailed recognition performs poorly on minority classes. The extremely imbalanced distribution of classifier weight norms leads to a decision boundary biased toward majority classes. To address this issue, we propose Class-Balanced Regularization to balance the distribution of classifier weight norms so that the model can make more balanced and reasonable classification decisions. In detail, CBR separately adjusts the regularization factors based on L2 regularization to be correlated with the class sample frequency positively, rather than using a fixed regularization factor. CBR trains balanced classifiers by increasing the L2 norm penalty for majority classes and reducing the penalty for minority classes. Since CBR is mainly used for classification adjustment instead of feature extraction, we adopt a two-stage training algorithm. In the first stage, the network with the traditional empirical risk minimization is trained, and in the second stage, CBR for classifier adjustment is applied. To validate the effectiveness of CBR, we perform extensive experiments on CIFAR10-LT, CIFAR100-LT, and ImageNet-LT datasets. The results demonstrate that CBR significantly improves performance by effectively balancing the distribution of classifier weight norms.

Keywords Imbalanced data · Long-tailed recognition · L2 Regularization · Decision boundary

1 Introduction

In recent years, deep learning has made incredible progress in computer vision [1–3]. It has been successfully applied to various visual discrimination tasks, including image classification [4], object detection [5], and semantic segmentation [6]. The widespread use of balanced datasets, such as Coco [7] and ImageNet [8], has largely contributed to the rapid development of deep learning. However, unlike these datasets, most real-world datasets often follow a long-tailed distribution, where majority classes occupy most of the data and minority classes have only a few samples [9, 10]. It is very challenging to train deep neural networks with imbalanced datasets. As shown in Fig. 1, with long-tailed datasets, deep neural networks generally perform well on majority classes and poorly on minority classes.

✉ Yuge Xu
xuyuge@scut.edu.cn

Chuanlong Lyu
luvchuanlong@gmail.com

[1] School of Automation Science and Engineering, South China University of Technology, Wushan Street, Guangzhou 510640, Guangdong, China

Published online: 29 April 2024

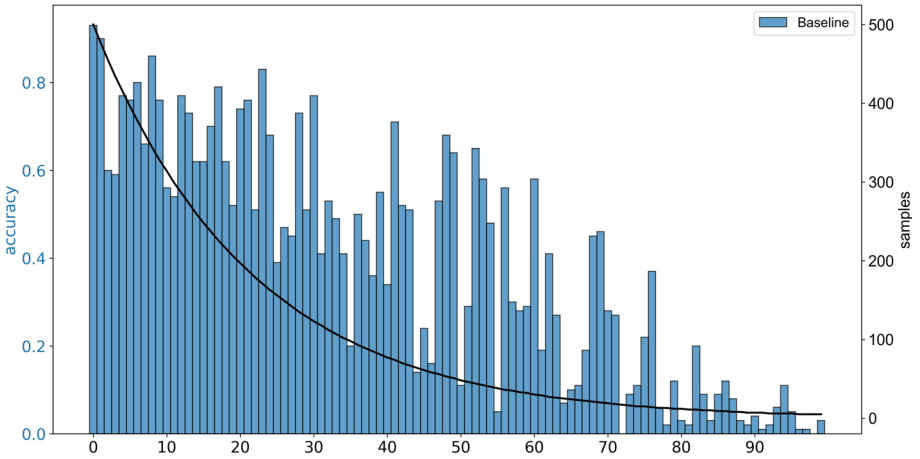

Fig. 1 The Top-1 accuracy (blue bars) of the model trained on CIFAR100-LT with an imbalance factor of 100. The black line shows the imbalanced class distribution of CIFAR100-LT. As observed, the model is biased toward the majority classes with many training samples

In order to address the long-tailed problem, some methods [11, 12] have been presented based on experimental observation. The traditional empirical risk minimization(ERM) approach yields the classifier with highly imbalanced weight norms. These norms are positively correlated with class sample frequencies as shown in Fig. 2. During the training process, the majority classes receive more positive gradients than minority classes due to the larger number of training samples [5]. As a result, the norms of classifier weights for the majority classes tend to increase, leading to an extreme imbalance in classifier weight norms. The imbalance compresses the decision space of the minority classes, which generally have small classifier weight norms. Consequently, networks tend to make decisions biased towards the majority classes while ignoring the samples from minority classes. Furthermore, according to the Decoupling study [11], training neural networks with a balanced dataset leads to classifier weights with similar norms. Therefore, balancing the distribution of classifier weight norms can help achieve a more balanced decision boundary and improve recognition performance.

To balance the distribution of the classifier weight norms, Decoupling [11] has proposed several methods, such as a post-processing scaling technique known as τ-normalization(τ-norm), and a training method called Classifier Re-training (cRT) which retrains the classifier with a balanced dataset by resampling. However, τ-norm merely scales the classifier weight norms. Due to the lack of corresponding adjustment to the classifier weight directions, τ-norm cannot balance the distribution of the classifier weight norms very well. cRT requires high extra training costs to train a relatively balanced classifier. Thus, both τ-norm and cRT have limitations in terms of effectively balancing the classifier. In this paper, our goal is to obtain a more balanced distribution of the classifier weight norms by adjusting the classifier with low extra training costs during the training process. A straightforward approach is to penalize the large classifier weight norms in training based on parameter regularization. Parameter regularization typically helps achieve a more balanced distribution of network weights to prevent over-fitting. Therefore, Weight Balancing [12] has investigated one parameter regularization method to balance network weights which is called Weight Decay. Weight balancing achieves higher performance by fine-tuning the weight decay factor, but it also fails to balance the classifier. For this issue, we argue that relying solely on a single identical regularization factor

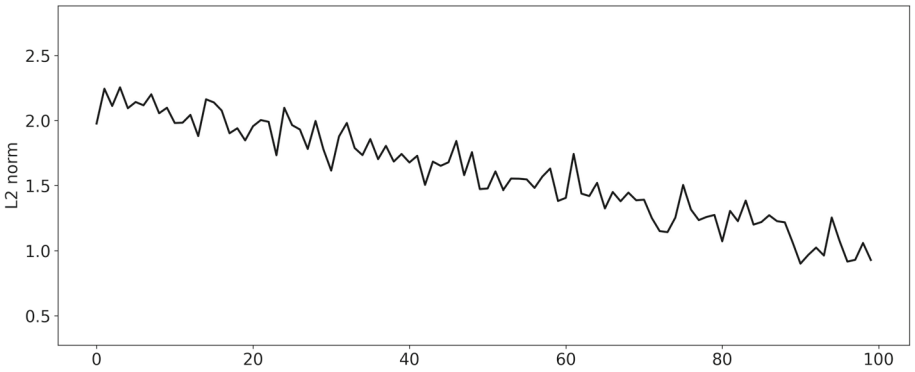

Fig. 2 The distribution of the norms of the classifier weights obtained by the traditional empirical risk minimization approach

cannot effectively balance the distribution of the classifier weight norms. The reason is that the positive gradients from the majority class samples are almost constant, while the regularization penalties decrease as the classifier weight norms decrease. Eventually, the majority classes with more samples will still have larger classifier weight norms.

In this paper, a novel technique called Class-Balanced Regularization (CBR) is proposed to improve long-tailed recognition performance by adjusting the classifiers. The method regulates the regularization factors separately for different classes to balance the distribution of the classifier weight norms. Specifically, CBR adjusts the L2 regularization factors for the classifier weights to ensure they are positively correlated with the class frequencies. Larger regularization factors are assigned to the majority classes while smaller ones are assigned to the minority classes. This results in stronger L2 norm penalties being imposed on the majority classes, which usually have larger classifier weight norms. As a result, CBR could achieve a more balanced distribution of the classifier weight norms. Moreover, the extra training cost for CBR is very low because it is applied after the well-trained imbalanced classifier.

The main contributions of our work can be summarized as follows:

1. We propose a novel technique called Class-Balanced Regularization (CBR) to adjust the imbalanced distribution of the classifier weight norms, aiming to achieve more balanced classification decisions of the recognition network.
2. We extend L2 regularization to balance the norms of the classifier weights, achieving significant performance improvement without high extra training costs.
3. We perform extensive experiments on various long-tailed datasets, including CIFAR10-LT, CIFAR100-LT, and ImageNet-LT. CBR consistently shows effective improvements and outperforms previous methods for balancing classifiers.

2 Related Work

2.1 Long-Tailed Recognition

In a long-tailed dataset, most minority classes have a small number of samples. The classification performance on these classes is poor, especially when the deep learning models are trained based on empirical risk minimization. Resampling techniques try to rebalance

the class distribution in the training data, including undersampling the majority classes [13, 14] and oversampling the minority classes [15, 16]. Reweighting techniques [4, 9, 17–19] assign different weights to the classes to emphasize the learning of the minority classes or to make the decision boundary of minority classes broader. In addition, reweighting methods [5, 20, 21] also include approaches that modify the gradients of training samples to achieve more balanced gradients for each class. In long-tailed learning, transfer learning [22–24] transfers the rich information from the majority classes to the minority classes, compensating for the limited information in the minority classes. Metric learning [25–27] seeks to explore distance-based losses to learn a more discriminative feature space. Invariant Feature Learning [28, 29] tries to focus on invariant features across different samples of the class. Ensemble learning [30–32] aims to improve overall performance by learning from more diverse distributions through multiple network branches, without sacrificing the performance of the majority classes. As a new paradigm in long-tailed learning, decoupled training [11] creatively decouples representation learning from classifier training and introduces various classifier adjustment methods, such as τ-norm and cRT. The Weight Balancing research [12] has explored one parameter regularization method called Weight Decay to enhance representation learning and balance classifiers.

2.2 Regularization

Regularization is indeed an important component in deep learning. It plays a crucial role in preventing over-fitting and improving generalization ability. One well-known regularization method is L2 regularization, which utilizes L2-norm penalties on the model parameters. There are other common regularization methods, such as Weight Decay [33], data augmentation [34], dropout [35], and early-stop strategy [36]. In this paper, we present Class-Balanced Regularization to alleviate the imbalance of the classifier weight norms based on L2 regularization.

3 Methods

3.1 Preliminaries

Assuming a long-tailed training set $D = \{(x_i, y_i)\}_i^N$, where each sample x_i is labeled as $y_i \in [1, ..., C]$. Let n_k denote the number of training samples for class k, and let $N = \sum_{k=1}^{C} n_k$ be the total number of training samples. The imbalance factor, $IF = \frac{n_{\max}}{n_{\min}}$ is usually utilized to measure the degree of imbalance in a long-tailed dataset.

In long-tailed recognition, the most commonly used classifier is the linear classifier, which calculates the output probabilities as follows:

$$p_k = \phi \left(w_k f_b(x_i) + b_k \right), \tag{1}$$

where ϕ denotes the softmax or sigmoid function, p_k is the predicted probability of the sample belonging to the k-th class. $f_b(x_i)$ is the feature representation of sample x_i output by the backbone network, w_k and b_k represent the classifier weight and bias corresponding to the k-th class, respectively. For simplicity, we ignore the bias term of the classifier which brings nearly no difference to the model performance. To train the network, a loss function usually is used to evaluate the error between the predicted probability p_k and the true label

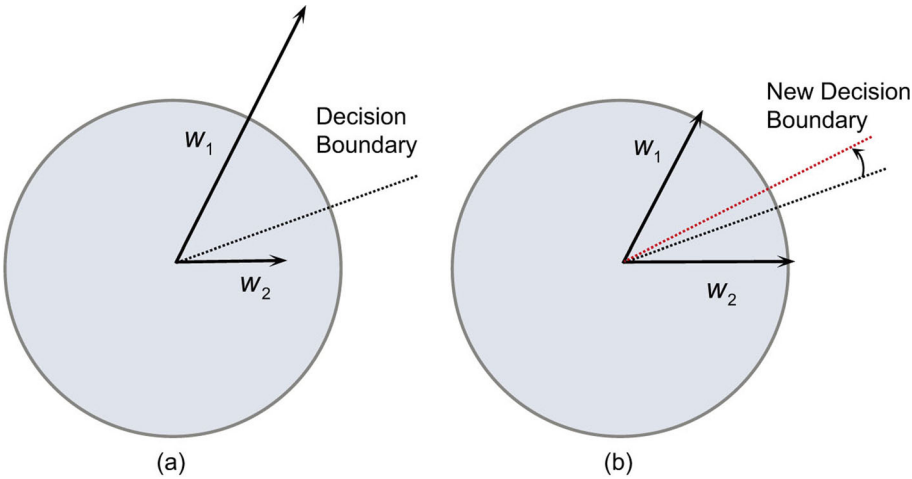

Fig. 3 Comparison of the decision boundary before and after using the classifier adjustment methods. **a** shows the imbalanced decision boundary obtained through training with long-tailed data. **b** illustrates the balanced decision boundary after adjusting the classifier weight norms

$\hat{p}_k$. The classification loss could be expressed as

$$L_{\mathrm{CE}} = \sum_{k=1}^{C} l\left(p_k, \hat{p}_k\right), \tag{2}$$

where l denotes a cross-entropy loss. In general, minimizing this classification loss is the training goal of recognition models. As shown in Fig. 1, the model trained on long-tailed data exhibits poor performance on minority classes. For the reasons behind this issue, Decoupling [11] claims that minimizing this loss guarantees high-quality representations in the learning model, but the long-tailed data mainly hurt the model's classification ability. In this study, we address the long-tailed problem by restoring the model's classification ability. In the following section, we analyze how imbalanced classifier weights weaken the classification performance.

3.2 The Problem in Long-Tailed Classification

To make correct classification decisions, the network should output the highest posterior probability corresponding to the ground truth class, which means $w_k f_b(x_i) > w_j f_b(x_i)$ for all classes $j \neq k$. However, in long-tailed recognition, the posterior probability of the ground truth class is often not the highest when the ground truth class is a minority class. The extremely imbalanced distribution of classifier weight norms leads to low classification accuracy. To explore how the distribution of classifier weight norms affects the decisions of the classifier, we consider the simplest binary classification problem with a decision boundary defined by

$$w_1 f_b(x_i) = w_2 f_b(x_i), \tag{3}$$

As shown in Fig. 3a, assuming there is a classifier with weights w_1 for class 1 and w_2 for class 2, and $\|w_1\| > \|w_2\|$, where $\|\bullet\|$ represents the L2 norm. The solid lines represent the

classifier weight vector, and the dashed lines indicate the classification boundary between the two classes. It can be observed that the decision boundary is close to class 2. Consequently, the decision space for class 2 is shrunk, while class 1 occupies a broader decision space. If the distribution of classifier weight norms is highly imbalanced i.e., $\|w_1\| \gg \|w_2\|$, the decision space of class 2 will be excessively compressed. The classifier will likely make biased classification decisions towards class 1. For the samples of class 2, they can only be classified correctly when the feature vector $f_b(x_i)$ and the weight vector w_2 are almost aligned. This leads to a serious challenge in classifying samples of class 2 correctly.

Multi-class classification can be considered as multiple binary classification problems. The analysis we discussed above can be generalized to the multi-class case. The majority classes of long-tailed datasets have a lot of samples. During the training process, the classifier weights corresponding to the majority classes tend to increase because of the numerous positive gradients. The distribution of the classifier weight norms will be highly imbalanced. The imbalanced classifier results in the decision boundary being biased towards the majority classes. From this perspective, a natural idea is to adjust the imbalanced classifier to balance the classification boundaries. After classifier adjustment, as Fig. 3b illustrates, the new decision boundary is equitable, allowing the network to make balanced classification decisions that are not biased to any specific class.

To adjust the imbalanced classifier, τ-norm is introduced in Decoupling [11]. The τ-norm method scales the classifier weights of a pre-trained model as follows:

$$\widetilde{w}_k = \frac{w_k}{\|w_k\|^\tau}, \tag{4}$$

where $\|\bullet\|$ represents the L2 norm, and w_k donates the classifier weight corresponding to the k-th class. $\widetilde{w}_k$ is the new classifier weight vector corresponding to class k after using the τ-norm method. τ is a hyper-parameter controlling the "temperature" of the normalization. The norm of w_k doesn't change if $\tau = 0$. As τ increases from 0 to 1, the differences between the classifier weight norms become progressively smaller. When $\tau = 1$, the Eq. 4 reduces to the standard post-hoc L2-normalization(L2-norm), which means all the norms of the classifier weights are equal to 1. Although the post-hoc L2-norm eliminates the imbalance in classifier weight norms, we believe that it is detrimental to the overall performance since it reduces the model's sensitivity to different classes. At the same time, we notice the other method, cRT, which can achieve excellent overall performance by retraining the classifier. Unfortunately, it requires much extra training cost with a restructuring dataset and couldn't effectively balance the distribution of the classifier weight norms. However, it inspires how to solve the classifier imbalance problem in training.

3.3 Class-Balanced Regularization

3.3.1 The Loss function

We expect to balance the classifier during training with less training cost, allowing the network to make more equitable classification decisions. A straight approach is to penalize large classifier weights, which is similar to commonly used parameter regularization. Parameter regularization is often employed during the training of deep neural networks, with L2 regularization being one of the most widely used techniques. L2 regularization applies the L2 norm penalty on network weights to prevent over-fitting and improve generalization. When L2 regularization is applied to the classifier weights, the classification loss function becomes

$$L_R = \sum_{k=1}^{C} l\left(p_k, \hat{p}_k\right) + \lambda \sum_k \|w_k\|^2, \tag{5}$$

where λ is the regularization factor, a hyper-parameter to control the strength of L2 regularization. The distribution of neural network weights could be influenced by modifying λ. A smaller regularization factor allows weights to adjust more freely, whereas a larger factor compels weights to shift toward smaller values. Equation 5 indicates that larger classifier weights are penalized more. Therefore, L2 regularization can mitigate the imbalanced weight norms to some extent, facilitating the learning of balanced network weights.

With the same thought, Weight Balancing [12] has investigated one parameter regularization method called Weight Decay. Weight Decay has the same effect as L2 regularization when the optimizer is SGD. Tuned Weight Decay(Tuned WD) based on Weight Decay searches for the optimal weight decay factor to solve the long-tailed problem. Tuned WD can achieve higher model performance by extracting high-quality feature representations but cannot balance the classifier. Same as Tuned WD, L2 regularization adopts a single identical regularization factor for each classifier weight vector. The single identical regularization factor may diminish the L2 regularization's ability to mitigate imbalance. The positive gradients from the samples of the majority classes are almost constant, but the regularization penalties decrease as the weight norms decrease. As a result, an equilibrium will be established between the positive gradients and the regularization penalties. Therefore, the majority classes with more samples will still have larger classifier weight norms. Hence, we propose a novel technique called Class-Balanced Regularization(CBR) to adjust the regularization factors separately for different classifier weight vectors. The classification loss is written as:

$$L_{\mathrm{CBR}} = \sum_{k=1}^{C} l\left(p_k, \hat{p}_k\right) + \sum_k \lambda_k \|w_k\|^2, \tag{6}$$

where λ_k is the regularization factor for the classifier weight vector corresponding to the k-th class. Compared to Eq. 5, the Eq. 6 has a more general form. When all the regularization factors have the same value, the Eq. 6 simplifies to the Eq. 5.

3.3.2 The Definition Strategies of the Regularization Factor

In long-tailed recognition, CBR should assign larger regularization factors to the majority classes. Therefore, the regularization factor to be positively correlated with the class sample frequency is defined as:

$$\lambda_k = \alpha \left(\frac{n_k}{n_{max}}\right)^{\beta}, \tag{7}$$

where n_k is the number of training samples for class k, and n_{max} is the maximum number of class samples. α represents the maximum strength of the L2 norm penalty, and β determines the distribution of the regularization factor λ_k. CBR aims to put larger L2 norm penalties on the classifier weights corresponding to the majority classes, encouraging the distribution of classifier weight norms to be more balanced. Therefore, the values of α and β should align with the imbalance degree of classifier weight norms, and together, they determine the actual balancing effect of CBR.

For the definition strategy of regularization factor λ_k, we also explored various forms. For example, we attempted to divide the classes into two or three splits based on the number of

class samples. The regularization factors are the same in a split and different among different splits. The specific forms are as follows:

$$\lambda_k = \begin{cases} \alpha, & n_k \geq N_1 \\ 0.01\alpha, & n_k < N_1 \end{cases} \tag{8}$$

$$\lambda_k = \begin{cases} \alpha, & n_k \geq N_2 \\ 0.01\alpha, & n_k \leq N_3 \\ 0.1\alpha, & \text{else} \end{cases}, \tag{9}$$

where N_1 is a threshold that divides the classes into majority and minority classes. A larger value indicates fewer classes are classified as majority classes. N_2, and N_3 are thresholds used to divide the classes into majority, medium, and minority classes. The accurate selection of these thresholds significantly influences the effectiveness of the method. Indeed, N_1 usually is set to the value that divides the classes into two splits equally. N_2 and N_3 are similarly used to divide the classes into three splits equivalently. Besides the forms mentioned above, we also utilize the dynamically adaptive regularization factors to balance the classifier, defined as Adaptive-Balanced Regularization (ABR):

$$\lambda_k = \alpha \left(\frac{\|w_k\| - \text{mean}\left(\|w_k\|\right)}{\text{std}\left(\|w_k\|\right)} \right), \tag{10}$$

where $mean()$ and $std()$ represent the mean and variance of the weight norms respectively. Same with the Eq. 7, α represents the maximum strength of the L2 norm penalty to the majority classes. ABR adjusts these factors at each iteration adaptively to achieve a more balanced distribution of the classifier weight norms. Compared to the above strategies, ABR has the least hyper-parameters. In the next section, ablation experiments were conducted to compare the performance of these different strategies.

3.4 Training Algorithm

The Decoupling study has found that long-tailed data allows the network to learn generalizable representations [11]. Additionally, the backbone network is shared among different classes and not directly related to the network's bias towards the majority classes. Thus, we focus on the imbalanced classifier in the study. To address the long-tailed problem, we primarily apply CBR to balance the distribution of the classifier weight norms. The proposed training algorithm consists of two stages: feature learning and classifier adjustment, as presented in Algorithm 1. The network is trained firstly by using ERM with the loss in Eq. 5 to learn a high-quality feature representation, and then the CBR loss in Eq. 6 is applied to adjust the imbalanced classifier. In the second stage of our training algorithm, there is no need for a balanced sampling of the training set or much additional training cost.

4 Experiments

In this section, we performed extensive experiments to evaluate the effectiveness of our proposed CBR technique. We first introduced the datasets, evaluation metrics, and implementation details, and then compared our method with previous long-tailed methods. We also conducted ablation experiments to analyze our CBR technique.

Algorithm 1 Class-Balanced Regularization Training Algorithm

Require: Long-tailed Dataset $D = \{(x_i, y_i)\}_i^N$, $y_i \in [1, ..., C]$.
Stage 1: Feature learning
Randomly initialize the model parameters θ_b and θ_c of backbone and classifier.
for $t = 1$ to T_0 **do**
  sample mini-batch D_m from D
  $L \leftarrow \frac{1}{m} \sum_{(x,y)\in D_m} L_R(y, f(x, \theta_b, \theta_c))$
  update model parameters θ_b and θ_c by minimizing L
end for
Stage 2: Classifier adjustment
for $t = T_0 + 1$ to T_1 **do**
  sample mini-batch D_m from D
  $L \leftarrow \frac{1}{m} \sum_{(x,y)\in D_m} L_{CBR}(y, f(x, \theta_b, \theta_c))$
  update model parameters θ_c by minimizing L
end for

4.1 Datasets and Evaluation Metric

To validate the effectiveness of our proposed CBR, extensive experiments were conducted on three widely used long-tailed datasets, i.e., CIFAR10-LT, CIFAR100-LT, and ImageNet-LT. These datasets are the long-tailed versions of CIFAR10 [37], CIFAR100 [37], and ImageNet2012 [8]. The CIFAR10 dataset includes 10 classes, each with 5000 training samples and 1000 test samples. The CIFAR100 dataset is similar to the CIFAR10, except it consists of 100 classes and each has 500 training samples and 100 test samples. The size of the image in CIFAR10 and CIFAR100 is 32×32. The ImageNet2012 dataset is a very frequently used dataset that consists of nearly 1.3 million images, with about 1300 training images and 50 validation images per class. We modified the balanced CIFAR10, CIFAR100, and ImageNet2012 datasets to generate the long-tailed versions by utilizing the exponential decay function $n = n_k \mu^k$, where n_k is the original number of training samples and $\mu \in (0, 1)$. We created three versions of train datasets for CIFAR10-LT and CIFAR100-LT respectively, and each with a different imbalance factor [10, 50, 100]. The imbalance factor of ImageNet-LT is 256. The most frequent class contains 1280 images, while the least has only 5 images. The validation sets of all datasets are balanced.

For each dataset, we trained models on the imbalanced training set and evaluated them on the balanced validation set. Top-1 classification accuracy is the main evaluation metric. To analyze the performance of these methods on the minority classes, the accuracy was further reported on three splits: Many-shot(≥ 100), Medium-shot($20 \sim 100$), and Few-shot (≤ 100) in addition to the overall accuracy.

4.2 Implementation Details

Our models were trained using the PyTorch toolbox [38]. For CIFAR10-LT and CIFAR100-LT, ResNet32 [1] was the baseline model and each model was trained for 200 epochs with a batch size of 128 at the first training stage. The SGD optimizer with a momentum of 0.9 was applied, and the initial learning rate was set to 0.1. The learning rate followed a linear warmup schedule in the first 5 epochs and decayed at the 160th and 180th epochs by 0.1.

Table 1 Top-1 accuracy on CIFAR10-LT and CIFAR100-LT with different imbalance factors [100, 50, 10]

Method	CIFAR100-LT			CIFAR10-LT		
	100	50	10	100	50	10
CE	38.32	43.85	55.71	70.36	74.81	86.39
CB-Focal [4]	39.60	45.17	57.99	74.57	79.27	87.10
LDAM-DRW [9]	42.04	46.62	58.71	77.03	81.03	88.16
BBN [30]	42.56	47.02	59.12	79.82	82.18	88.32
Remix [22]	41.94	–	59.36	75.36	–	88.15
CMO [24]	43.9	48.3	59.5	–	–	–
TSC [26]	43.8	47.4	59.0	79.7	82.9	88.7
De-confound [41]	44.1	50.3	59.6	**80.6**	**83.6**	88.5
IBLoss [18]	42.14	46.22	57.13	78.26	81.70	88.25
cRT [11]‡	45.54	50.20	60.61	77.28	82.36	87.79
τ-norm [11]‡	45.83	49.79	60.45	79.48	82.01	88.27
CBR	**47.13**	**50.63**	**61.01**	80.36	82.53	**89.02**

‡ denotes our reproduced results. The best results are marked in bold

For ImageNet-LT, ResNeXt50 [39] was the baseline model. Each model was trained for 90 epochs with a batch size of 128 at the first training stage. The optimizer was also SGD with an initial learning rate of 0.05. The learning rate followed a cosine annealing scheduler. For the second stage, models were trained using CBR for 5 epochs and other classifier adjustment methods for 10 epochs. Random horizontal flipping and cropping were implemented as simple augmentation. We utilized grid search to optimize hyper-parameters.

4.3 Results and Discussion

To validate the effectiveness of the proposed method, we conducted extensive experiments and compared it with different types of long-tailed methods, such as Focal [17], CB [4], and IBLoss [18] for loss reweighting; Remix [22] and CMO [24]for data mixing augmentation; LDAM [9], Logit adjustment [19], and ALALoss [40] for logit modification; TSC [26] for targeted supervised contrastive learning. De-confound [41] uses a causal classifier. BBN [30] applies two network branches to learn representation and train the classifier respectively. τ-norm [11], cRT [11], and Tuned WD [12] are typical classifier adjustment methods. CE means the network trained with ERM. If not otherwise specified, the data of other methods in the tables are from the reported results in the original paper.

4.3.1 Results on CIFAR10-LT and CIFAR100-LT.

We performed extensive experiments on CIFAR10-LT and CIFAR100-LT datasets to compare CBR with previous long-tailed methods. Table 1 reports the classification accuracy of various methods on CIFAR10-LT and CIFAR100-LT datasets. It can be seen that CBR outperforms most of the shown methods. In detail, CBR achieves the best classification results on the CIFAR100-LT datasets with imbalance factors of 10, 50, and 100, and the results are 47.13%, 50.63%, and 61.01%. CBR surpasses the advanced benchmark CMO [24] by 3.23%, 1.33%, and 1.51% respectively. According to Table 1, CBR attains the best performance on the CIFAR10-LT dataset with an imbalance factor of 10. The classification accuracies of CBR

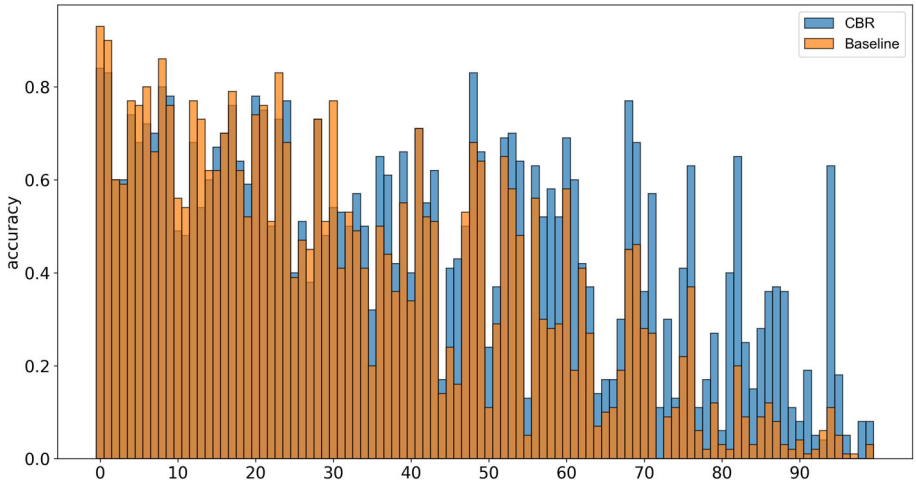

Fig. 4 Comparison of the Top-1 accuracy on CIFAR100-LT(IF=100) of the networks before and after using CBR. Orange bars: the baseline network. Blue bars: the network which has a more significant performance by applying CBR

are slightly lower than De-confound by 0.24% and 1.07% on CIFAR10-LT datasets with imbalance factors of 100 and 50 respectively. We believe that the more the number of classes in the long-tailed datasets, the more serious the impact of the imbalance of the classifier weight norms. Compared to the other two classifier adjustment methods, cRT [11] and τ-norm [11], CBR has superior performance on both CIFAR10-LT and CIFAR100-LT datasets.

Figure 4 presents a comparison of class-wise classification accuracy between the model using CBR and the baseline model trained on the CIFAR100-LT dataset with an imbalance factor of 100. CBR significantly improves the classification accuracy of the minority classes. In some cases, the performance increases from less than 10% to over 60%. It is worth noting that CBR slightly reduces the performance of a few majority classes. The excellent performance of majority classes is not only due to their abundant samples but also because of the compression of the decision space for minority classes. Overall, CBR greatly improves the model's performance on long-tailed data. The distributions of classifier weight norms before and after applying the CBR method are shown in Fig. 5. After applying the CBR method, the distribution of classifier weight norms is more balanced compared to the originally imbalanced classifier.

4.3.2 Results on ImageNet-LT

To present CBR's performance on large-scale datasets, we evaluated CBR on ImageNet-LT. In addition to the overall accuracy, the results on Many-shot, Medium-shot, and Few-shot groups are reported in Table 2. As can be seen, CBR achieves the best performance in overall accuracy i.e., 53.87%. More specifically, CBR outperforms DisAlign [42], De-confound [41], and ALALoss [40] in terms of overall performance by 1.27%, 2.07%, and 0.57%, respectively. Similarly, CBR also exhibits better performance than the other classifier adjustment methods such as cRT, τ-norm, and Tuned WD. Besides, CBR gains significant improvements in the performance of Medium-shot and Few-shot classes without excessively sacrificing the

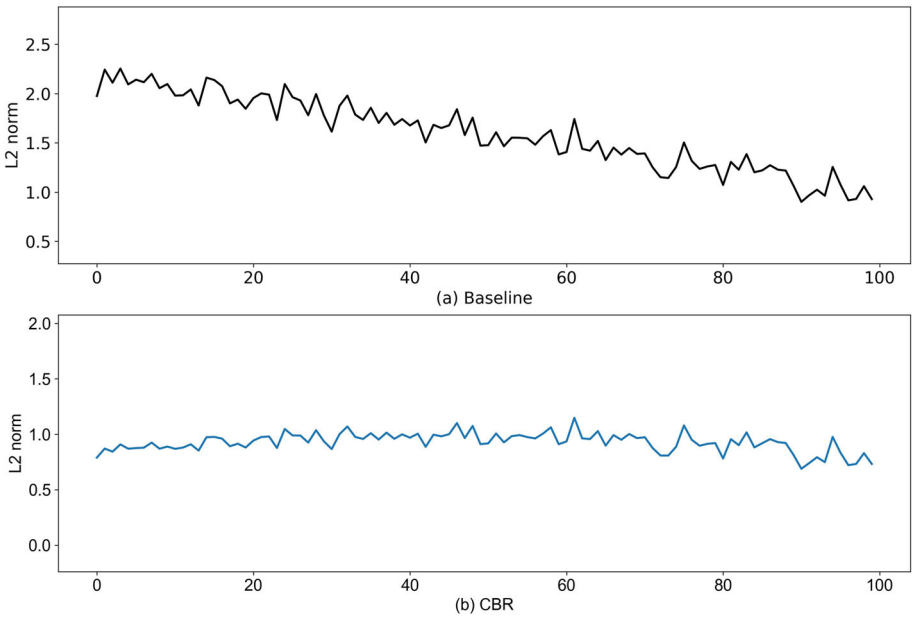

Fig. 5 Comparison of the distributions of the classifier weight norms before and after using CBR. **a** The highly imbalanced weight norm distribution of the original classifier. **b** The classifier becomes more balanced after applying our CBR method

Table 2 Top-1 accuracy on ImageNet-LT dataset

Method	Many	Medium	Few	All
CE	**65.9**	37.5	7.7	44.4
OLTR [10]	–	–	–	46.3
Logit adjustment [19]	–	–	–	51.11
DisAlign [42]	61.5	50.7	33.1	52.6
De-confound [41]	62.7	48.8	31.6	51.8
ALALoss [40]	64.1	49.9	34.7	53.3
cRT [11]	61.8	46.2	27.4	49.6
τ-norm [11]	59.1	46.9	30.3	49.4
Tuned WD [12]	62.0	49.7	**41.0**	53.3
CBR	62.60	**52.08**	35.44	**53.87**

The best results are marked in bold

performance of majority classes. This demonstrates that CBR can effectively balance the classifier and obtain superior overall performance on the ImageNet-LT dataset.

4.4 Ablation Study

To further analyze our proposed CBR, we performed several ablation studies. All experiments were conducted on the CIFAR100-LT dataset with an imbalance factor of 100.

Table 3 Ablation studies of different classifier adjustment methods on CIFAR100-LT(IF=100)

Method	Many	Medium	Few	All
Baseline network	**71.40**	41.71	8.07	42.01
+L2-norm	56.00	41.40	**33.47**	44.13
+τ-norm	64.71	45.49	24.20	45.83
+cRT	65.91	43.74	23.87	45.54
+CB&Tuned WD	66.23	45.09	24.00	46.16
+CBR	60.54	**51.03**	26.93	**47.13**

We used the methods after "+" to adjust the classifier in the second stage. The best results are marked in bold

4.4.1 Different Classifier Adjustment Methods

The training process is divided into two stages. The first stage is feature learning, where the network is trained by using ERM with L2 regularization. The second stage is the classifier adjustment, where methods are utilized to adjust the classifier. We chose several commonly used classifier adjustment methods for comparison, including L2-norm, τ-norm, cRT, and Tuned WD. As shown in Table 3, CBR achieves the best result with an overall accuracy of 47.13% among these methods. Although the L2-norm performs the best in the Few-shot split, its significant drop in the Many-shot split leads to no notable overall improvement. This suggests that simply post-processing the classifier weight norms to make them entirely identical is not an effective approach for tackling the long-tailed problem. Furthermore, CBR outperforms the τ-norm method, indicating that the imbalance issue in the classifier may not be just the imbalance in weight norms, which τ-norm fails to address. Compared to cRT and Tuned WD, CBR achieves higher performance and only requires 5 epochs to mitigate the imbalance in the classifier, which demonstrates that CBR only needs less additional training cost to balance the classifier more significantly.

4.4.2 Different Strategies of Regularization Factor

To further analyze our proposed CBR, we also explored several different strategies for the definition of the regularization factor λ_k, including fixed and adaptive strategies which can be seen in Table 4. The fixed strategies include Two-Part, Three-Part, and Exponential strategies. The adaptive strategy is the ABR method that we discuss in Sect. 3.3.2. The Exponential strategy attains the best results not only in overall accuracy but also in accuracy on the medium and few splits in Table 4. The Two-part and Three-part strategies only exhibit marginal improvements because of their limited ability to resolve internal imbalances within the splits. ABR shows comparable performance to τ-norm and cRT, without adding any extra hyper-parameters. Therefore, we believe that ABR is also a considerable approach.

4.4.3 The Difference Between CBR and Tuned Weight Decay

Tuned WD [12] and CBR both address the long-tailed problem from the perspective of regularization. However, Weight Decay trims network parameters by a certain proportion after each iteration, while L2 regularization alters the optimization objective to impose constraints on the norms of network parameters. Weight Decay and L2 regularization have a similar effect only when the optimizer is SGD. Tuned WD aims to search for the optimal weight

Table 4 Ablation studies of different strategies for the regularization factor on CIFAR100-LT(IF=100)

Strategy	λ_k	Many	Medium	Few	All
Two-Part	$\begin{cases}\alpha & n_k \geq N_1 \\ 0.01\alpha & n_k < N_1\end{cases}$	64.23	50.31	15.13	44.63
Three-Part	$\begin{cases}\alpha & n_k \geq N_2 \\ 0.01\alpha & n_k \leq N_3 \\ 0.1\alpha & \text{else}\end{cases}$	63.06	50.11	17.4	44.83
ABR	$\alpha\left(\frac{\|w_k\|-\text{mean}(\|w_k\|)}{\text{std}(\|w_k\|)}\right)$	**64.74**	44.83	24.97	45.84
Exponential	$\alpha\left(\frac{n_k}{n_{max}}\right)^{\beta}$	60.54	**51.03**	**26.93**	**47.13**

The best results are marked in bold

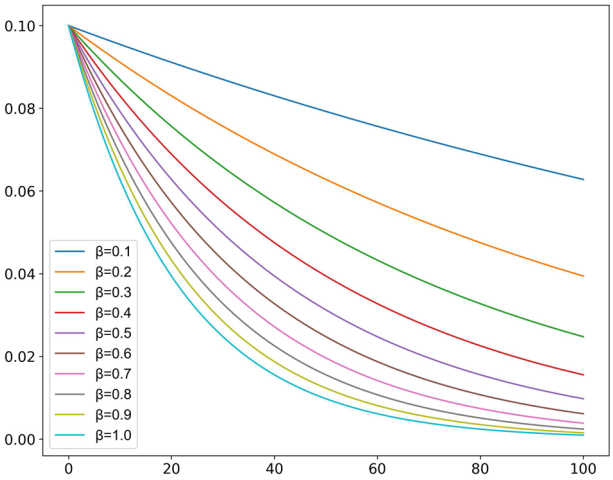

Fig. 6 The distributions of the regularization factor varying with different β when $\alpha = 0.1$ on CIFAR100-LT(IF=100). As β increases progressively, the distribution becomes steeper

decay factor to facilitate representation learning. However, when used independently, it shows limited ability in balancing classifiers. In contrast, our CBR adjusts the regularization factor individually to balance the distribution of the classifier weight norms, applying different L2 norm penalties on the classifier weights corresponding to different classes. Thus, from the conceptual and practical perspectives, Tuned WD and CBR are both different methods. As shown in Table 3, in the second training stage, Tuned WD usually needs to be combined with reweighting methods to achieve better performance. On the other hand, CBR outperforms without requiring modifications to the training set or loss function.

4.4.4 The Analysis of Hyper-Parameters

In Eq. 7, α represents the maximum strength of the L2 norm penalty to the majority classes, while β determines the distribution of the regularization factor λ_k. Figure 6 depicts the distributions of the regularization factor with different β when $\alpha = 0.1$ on CIFAR100-LT with an imbalance factor of 100. As β increases progressively, the distribution becomes

Table 5 Ablation studies of impact of the CBR hyper-parameters on CIFAR100-LT(IF=100)

α	β	Many	Medium	Few	All
0.2	1	61.12	51.80	21.90	46.12
0.25	1	62.00	52.86	20.20	46.26
0.3	1	58.86	54.00	22.63	46.29
0.35	1	55.06	**54.94**	24.77	45.93
0.3	0.2	**66.71**	47.29	20.97	46.19
0.3	0.3	63.17	49.34	24.77	46.81
0.3	0.4	60.54	51.03	26.93	**47.13**
0.3	0.5	58.71	52.29	**27.07**	46.97
0.3	0.6	57.91	53.31	26.27	46.81

The best results are marked in bold

Table 6 The optimal values of α and β for different datasets

Dataset	CIFAR100-LT			CIFAR10-LT			ImageNet-LT
Imbalance factor	100	50	10	100	50	10	256
α	0.3	0.15	0.1	0.6	0.4	0.15	0.15
β	0.4	0.6	0.9	1.0	1.0	1.0	0.4

steeper. The regularization factors for the majority classes are approximately equal to α, while the factors for the minority classes are small.

To analyze the effects of α and β, experiments on the CIFAR100-LT(IF=100) with different parameter sets were conducted as shown in Table 5. We initially adjusted α using grid search while keeping β fixed at 1 and subsequently fine-tuned β. As shown in Table 5, the best performance is achieved when α and β are set to 0.3 and 0.4, respectively. Although there is a slight decrease in performance when modifying the parameter set, the results remain at an outstanding level. When α or β increases, there is an obvious increasing trend of accuracy in the medium-shot and few-shot classes, whereas accuracy tends to decline in the many-shot classes.

The optimal parameter sets for each dataset are summarized in Table 6. The optimal values of α and β are dataset-dependent. In the case of CIFAR100-LT and CIFAR10-LT, the optimal value of α increases as the imbalance factor increases. This suggests that when the class imbalance degree increases, a larger L2 norm penalty for the majority class is required to balance the classifier. The optimal value of β for CIFAR100-LT decreases as the imbalance level increases, while the optimal value of β for CIFAR10-LT remains consistently at 1.0. This indicates that the optimal value of β may be related to the number of classes and the class imbalance degree in the dataset.

5 Conclusion

In order to solve the class imbalance problem in long-tailed recognition, we propose a novel technique called Class-Balancing Regularization (CBR), which separately adjusts the L2 regularization factors of different classifier weights based on the class sample frequencies. CBR applies larger L2 norm penalties to the majority classes for adjusting the classifier during training. We also adopt a two-stage training algorithm to utilize the CBR method.

Extensive experiments are conducted on various long-tailed datasets, and the results consistently demonstrate the effectiveness of CBR. The proposed CBR significantly balances the distribution of the classifier weight norms and outperforms previous long-tailed methods. In future work, we aim to study how to combine our method with existing techniques to handle long-tailed problems further.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Las Vegas, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
2. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
3. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement https://doi.org/10.48550/arXiv.1804.02767
4. Cui Y, Jia M, Lin T-Y, Song Y, Belongie S (2019) Class-balanced loss based on effective number of samples. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Long Beach. https://doi.org/10.1109/cvpr.2019.00949
5. Tan J, Wang C, Li B, Li Q, Ouyang W, Yin C, Yan J (2020) Equalization loss for long-tailed object recognition. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Seattle, pp. 11659–11668. https://doi.org/10.1109/CVPR42600.2020.01168
6. Wang J, Zhang W, Zang Y, Cao Y, Pang J, Gong T, Chen K, Liu Z, Loy CC, Lin D (2021) Seesaw loss for long-tailed instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9695–9704
7. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In: Computer vision–ECCV 2014. Springer, Zurich, pp. 740–755
8. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, Miami, pp. 248–255
9. Cao K, Wei C, Gaidon A, Arechiga N, Ma T (2019) Learning imbalanced datasets with label-distribution-aware margin loss. In: Advances in neural information processing systems. Curran Associates, Inc., Vancouver, vol 32
10. Liu Z, Miao Z, Zhan X, Wang J, Gong B, Yu SX (2019) Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2537–2546
11. Kang B, Xie S, Rohrbach M, Yan Z, Gordo A, Feng J, Kalantidis Y (2020) Decoupling representation and classifier for long-tailed recognition. In: International conference on learning representations
12. Alshammari S, Wang Y-X, Ramanan D, Kong S (2022) Long-tailed recognition via weight balancing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6897–6907
13. Estabrooks A, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced data sets. Comput Intell 20(1):18–36. https://doi.org/10.1111/j.0824-7935.2004.t01-1-00228.x

14. Liu X-Y, Wu J, Zhou Z-H (2008) Exploratory undersampling for class-imbalance learning. IEEE Trans Syst Man Cybern Part B (Cybern) 39(2):539–550
15. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J. Artif Intell Res 16:321–357. https://doi.org/10.1613/jair.953
16. Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Advances in intelligent computing: international conference on intelligent computing. ICIC 2005. Springer, Hefei, pp 878–887
17. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
18. Park S, Lim J, Jeon Y, Choi JY (2021) Influence-balanced loss for imbalanced visual classification. In: 2021 IEEE/CVF international conference on computer vision (ICCV). IEEE, Montreal, pp 715–724. https://doi.org/10.1109/ICCV48922.2021.00077
19. Menon AK, Jayasumana S, Rawat AS, Jain H, Veit A, Kumar S (2021) Long-tail learning via logit adjustment. In: International conference on learning representations
20. Tan J, Lu X, Zhang G, Yin C, Li Q (2021) Equalization loss v2: a new gradient balance approach for long-tailed object detection. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Nashville, pp 1685–1694. https://doi.org/10.1109/CVPR46437.2021.00173
21. Wang T, Zhu Y, Zhao C, Zeng W, Wang J, Tang M (2021) Adaptive class suppression loss for long-tail object detection. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Nashville, pp 3102–3111. https://doi.org/10.1109/CVPR46437.2021.00312
22. Chou H-P, Chang S-C, Pan J-Y, Wei W, Juan D-C (2020) Remix: Rebalanced mixup. In: Bartoli A, Fusiello A (eds) Computer Vision–ECCV 2020 Workshops, vol 12540. Springer, Cham, pp 95–110. https://doi.org/10.1007/978-3-030-65414-6_9
23. Kim J, Jeong J, Shin J (2020) M2m: imbalanced classification via major-to-minor translation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13896–13905
24. Park S, Hong Y, Heo B, Yun S, Choi JY (2022) The majority can help the minority: context-rich minority oversampling for long-tailed classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6887–6896
25. Huang C, Li Y, Loy CC, Tang X (2016) Learning deep representation for imbalanced classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5375–5384
26. Li T, Cao P, Yuan Y, Fan L, Yang Y, Feris RS, Indyk P, Katabi D (2022) Targeted supervised contrastive learning for long-tailed recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6918–6928
27. Pang Z, Wang C, Wang J, Zhao L (2023) Reliability modeling and contrastive learning for unsupervised person re-identification. Knowl Based Syst 263:110263. https://doi.org/10.1016/j.knosys.2023.110263
28. Tang K, Tao M, Qi J, Liu Z, Zhang H (2022) Invariant feature learning for generalized long-tailed classification. In: Avidan S, Brostow G, Cissé M, Farinella GM, Hassner T (eds) Computer vision–ECCV. 2022 lecture notes in computer science. Springer, Cham, pp 709–726. https://doi.org/10.1007/978-3-031-20053-3_41
29. Pang Z, Zhao L, Liu Q, Wang C (2023) Camera invariant feature learning for unsupervised person re-identification. IEEE Trans Multimed 25:6171–6182. https://doi.org/10.1109/TMM.2022.3206662
30. Zhou B, Cui Q, Wei X-S, Chen Z-M (2020) BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Seattle, pp. 9716–9725. https://doi.org/10.1109/CVPR42600.2020.00974
31. Xiang L, Ding G, Han J (2020) Learning from multiple experts: self-paced knowledge distillation for long-tailed classification. In: Vedaldi A, Bischof H, Brox T, Frahm J-M (eds) Computer vision–ECCV. 2020 lecture notes in computer science. Springer, Cham, pp 247–263. https://doi.org/10.1007/978-3-030-58558-7_15
32. Cai J, Wang Y, Hwang J-N (2021) ACE: ally complementary experts for solving long-tailed recognition in one-shot. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 112–121
33. Krogh A, Hertz J (1991) A simple weight decay can improve generalization. In: Advances in neural information processing systems, vol. 4. Morgan-Kaufmann, Denver, pp. 950–957
34. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):1–48
35. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
36. Prechelt L (2012) Early Stopping–but When? Neural Networks: Tricks of the Trade. Springer, Heidelberg, pp 53–67
37. Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images

38. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch
39. Xie S, Girshick R, Dollar P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492–1500
40. Zhao Y, Chen W, Tan X, Huang K, Zhu J (2022) Adaptive logit adjustment loss for long-tailed visual recognition. In: Thirty-sixth AAAI conference on artificial intelligence/thirty-fourth conference on innovative applications of artificial intelligence/the Twelveth symposium on educational advances in artificial intelligence. The Association Advancement Artificial Intelligence, Palo Alto, pp. 3472–3480
41. Tang K, Huang J, Zhang H (2020) Long-tailed classification by keeping the good and removing the bad momentum causal effect. In: Advances in neural information processing systems, vol. 33. Curran Associates, Inc., Virtual, pp. 1513–1524
42. Zhang S, Li Z, Yan S, He X, Sun J (2021) Distribution alignment: a unified framework for long-tail visual recognition. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Nashville, pp. 2361–2370. https://doi.org/10.1109/CVPR46437.2021.00239