Pattern Analysis and Applications (2025) 28:30
https://doi.org/10.1007/s10044-025-01410-3

ORIGINAL ARTICLE

Balanced clustering contrastive learning for long-tailed visual recognition

Byeong-il Kim[1] · **Byoung Chul Ko**[1]

Received: 29 June 2024 / Accepted: 1 January 2025

Abstract
Real-world deep learning training data often follow a long-tailed (LT) distribution, where a few classes (head classes) have the most samples and many classes (tail classes) have very few samples. Models trained on LT datasets typically achieve high accuracy on head classes, but suffer from poor performance on tail classes. To address this challenge, strategies based on supervised contrastive learning have been explored. However, existing methods often focus on either reducing the dominance of head class features or expanding the feature space of tail classes, but rarely achieve a balanced feature distribution across both. In this paper, we propose Balanced clustering contrastive learning (BCCL) to balance the feature space between the head and tail classes more effectively. The proposed approach introduces two main components. First, we employ queue-based clustering to extract multiple centroids. This addresses the intra-minibatch class absence issue and maintains intra-class balance. Second, we expand the feature space of tail classes based on class frequency to enhance their expressiveness. An evaluation of four LT datasets, CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018, demonstrates that BCCL consistently outperforms the existing methods. These results establish the ability of BCCL to maintain a balanced feature space in diverse environments. Our code is available at https://github.com/GGTINE/BCCL.

Keywords Long-tailed visual recognition · Supervised contrastive learning · Queue-based clustering · Feature space balancing · Logit compensation

1 Introduction

Advancements in deep learning and the emergence of large-scale datasets have significantly affected the field of computer vision, particularly in tasks such as image classification [1–3], object detection [4], and segmentation [5]. A critical factor contributing to these advancements is the availability of datasets that exhibit rich diversity and balanced distributions. High-quality datasets play a pivotal role in training deep models to generalize effectively across various scenarios. Rich datasets expose models to a wide range of objects, backgrounds, and lighting conditions, thereby ensuring robust performance in real-world applications. Moreover, balanced data distributions ensure that the models learn equitably from all classes, thus mitigating biases towards any particular class. However, the construction of comprehensive datasets that maintain diversity while representing real-world distributions remains a formidable challenge. In practice, most deep-learning training datasets exhibit long-tailed (LT) distributions. This distribution is characterised by a few classes that dominate the dataset with numerous samples, whereas the majority of classes have only a few samples. For example, the ImageNet dataset [6] includes various object categories; however, only a few of these categories contain a large number of images, whereas most categories are underrepresented. In LT-distributed datasets, deep models tend to achieve high accuracy on the head classes (classes with abundant samples), but perform poorly on the tail classes (classes with few samples). This discrepancy can lead to models that fail to generalize well to real-world data and produce biased outcomes for certain classes. Consequently, effective learning from LT-distributed datasets remains a critical challenge in the field of computer vision.

✉ Byoung Chul Ko
niceko@kmu.ac.kr

Byeong-il Kim
wqpo1235@gmail.com

[1] Department of Computer Engineering, Keimyung University, Daegu 42601, South Korea

Published online: 15 January 2025

To address the issue of model bias in LT-distributed datasets, various strategies have been explored. Class rebalancing techniques [7–11] adjust the weights or sampling ratios during training to account for the class distribution. These methods transfer rich information from the head classes to readjust decision boundaries via transfer learning, thereby compensating for the scarcity of samples in the tail classes. Recent studies on Supervised contrastive learning (SCL) [12] have proposed training methods that bring samples of the same class closer in feature space while pushing samples of different classes apart. This approach effectively learns robust representations even with limited data, making it suitable for learning from LT distributions. In particular, algorithms that focus on redesigning a biased feature space by concentrating on the head classes [13–15] have often outperformed competing algorithms. However, these methods primarily reallocate the feature space by shrinking the feature space of the head classes to adjust the feature space of the tail classes; however, they do not make any direct attempt to expand the limited feature space of the tail classes caused by the imbalanced distribution. Therefore, to enhance both the fairness and performance of models trained on LT-distributed datasets, new approaches that directly expand the feature space of the tail classes are necessary.

Therefore, we propose a method called Balanced clustering contrastive learning (BCCL). Figure 1 provides a visual explanation of how BCCL adjusts a biased feature space to construct a balanced feature space. The proposed BCCL leverages the advantages of SCL while introducing a novel strategy to balance the feature space between the head and tail classes in LT distributions more effectively. Unlike existing approaches, which primarily compress the feature space of the head classes, our approach aims to enable deep models to achieve balanced learning in LT distributions by expanding the feature space of the tail classes through clustering.

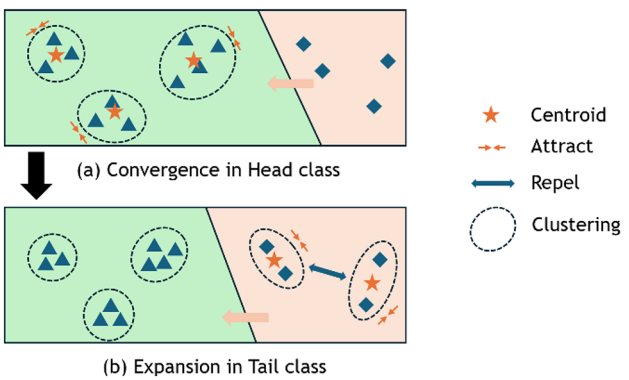

Fig. 1 Feature space adjustment using BCCL. In **a**, the feature points (blue triangles) of the head class converge towards the centroids (orange stars) due to strong attractive forces, resulting in compact clustering and clear inter-class separation. In **b**, repulsive forces from the centroids cause the feature points (blue diamonds) of the tail class to spread out, thus overcoming the narrow feature space. This balances the feature space by compacting the head classes and expanding the tail classes (Color figure online)

BCCL addresses the problem caused by the LT distribution by introducing the following two key components: (1) queue-based clustering and (2) feature space compression and expansion. In the first component, queue-based clustering is used to extract multiple centroids. These multiple centroids mitigate the issue of class omission, which commonly occurs within the mini-batches of LT distributions and maintains both interclass and intraclass balance using a centroid loss function. In the second component, to enable the model to adapt to the data distribution, the feature space of the head classes is converged while that of the tail classes is expanded according to the class frequencies. This enhancement improves the underrepresentation of the tail classes caused by the bias in an LT distribution. This approach alleviates the class imbalance problem by maintaining a balanced feature space for both types of classes.

The proposed BCCL method was evaluated on four benchmark datasets with LT distributions (CIFAR10-LT [16], CIFAR100-LT [16], ImageNet-LT [17], and iNaturalist 2018 [18]) to assess the generalization capability of deep models in image classification. Through experimental results, we demonstrated that the proposed method consistently outperforms existing competitive LT recognition methods and maintains a balanced feature space across various settings. The contributions of this study are as follows:

- We propose a queue-based multiple clustering method and a feature space contraction and expansion approach that adapts to the data distribution to mitigate the bias of deep models. This effectively balances the classes with different distributions.
- We introduce a centroid loss that leverages multiple centroids to achieve balanced feature space distributions across classes. Additionally, a class-complement strategy addresses class imbalance within mini-batches during training, ensuring fair representation of all classes in LT distributions.
- Through experiments, we demonstrate that the proposed BCCL achieves superior performance on various LT benchmark datasets while maintaining a balanced feature space, thereby proving its effectiveness in addressing the data imbalance problem in LT-distributed datasets.

2 Related works

LT Recognition has been actively researched not only in class recognition but also in various machine learning fields to mitigate the impact of imbalanced data distributions. Class re-balancing [7, 10, 11, 19–22] aims to modify the training distribution to reduce imbalance. However, this method may lead to information loss by randomly discarding samples (under-sampling) for the majority classes, while causing overfitting for the minority classes due to repeated samples (over-sampling). Deep-RTC [7] enhanced the recognition performance of tail classes by utilizing probabilistic tree sampling during training and rejection mechanisms during inference. MIMB [23] is designed to reduce imbalances within datasets by integrating advanced data recovery techniques with consistency exploration through reverse regularization, ensuring a more robust and accurate representation of incomplete multi-view data. Breadcrumbs [11] focused on tail class recognition during the learning process. At each epoch, features generated during training are back-tracked to perform data augmentation based on features generated in previous epochs. This increases the amount of data for tail classes and helps the model learn tail classes more effectively. Representation Learning [15, 24–30] redesigned the feature space to enhance the generalization performance of tail classes. By emphasizing the distinctiveness between tail and head classes in the feature space design, the model is assisted in learning tail classes more effectively.

Contrastive Learning for LT recognition. Contrastive learning [31, 32] is a self-supervised learning approach that utilizes a contrastive loss function to maximize the similarity between positive samples and minimize the similarity between positive and negative samples, thereby learning a more discriminative representation of the data. Parametric contrastive learning (PaCo) [33] addressed this issue by introducing class-wise learnable centers that rebalance the optimization landscape. To tackle the issue of LT classes clustering with similar ones, Adaptive hierarchical representation learning (AHRL) [34] proposed a hierarchical partitioning of the entire feature space and addresses the problem in a coarse-to-fine manner. Joint representation and classifier learning (JRCL) [35] proposed an algorithm that jointly optimizes a supervised contrastive loss, a binary distribution consistency loss, and a multi-classification loss. Ma et al. [36] defined a metric for measuring the geometric structure of feature distributions and the similarity between them and proposed a feature uncertainty representation that utilizes the geometric information of the head class feature distribution to perturb the tail class features.

Balanced contrastive learning (BCL) [13] proposed a loss function for balanced contrastive learning to further improve the visual recognition performance of LT. BCL incorporates class balancing, which balances the gradient contributions from negative classes, and class-complement, which ensures that all classes are present in every mini-batch. Feature clusters compression (FCC) [15] proposed to increase the density of backbone features by compressing backbone feature clusters. Subclass balanced contrast learning (SBCL) [14] captures the two-level class hierarchy between the original class and its subclasses. To achieve this, each head class is clustered into several subclasses of similar size as the tail class, and the learned representation is trained to understand this two-level class hierarchy. Du et al. [37] proposed a Probabilistic contrastive (ProCo) learning algorithm that estimates the sample data distribution of each class in the feature space and samples contrastive pairs accordingly. However, existing tail class-centric strategies based on contrastive learning fail to capture class-specific characteristics, thus limiting their performance improvement.

Data augmentation for LT recognition [38–42] is an effective technique for increasing the representation of limited tail class samples, thereby reducing overfitting and enhancing the representativeness of the data. CUDA [38] proposed a method to overcome these limitations by adjusting the augmentation intensity based on the class-wise model prediction accuracy. Through Level-of-Learning (LoL) evaluation, it assesses the learning levels of each class and adjusts the augmentation intensity accordingly. For classes with high LoL, more challenging data are provided, while for classes with low LoL, data are adjusted to facilitate easier learning, allowing each class to be trained in its optimal learning environment. SAFA [39] addresses the problem of tail class data augmentation not generalizing during the testing phase by extracting diverse transferable semantic directions from head classes to adaptively transform the features of tail classes.

This study tackles the challenges of LT distributions by proposing a novel approach that balances feature distributions through the contraction of head class feature spaces and the expansion of tail class feature spaces. By concurrently reducing the dominance of head class features and enhancing the representation of tail classes, the proposed method achieves a more equitable and comprehensive feature distribution.

3 Method

3.1 Preliminaries

In this section, we introduce the key concepts of logit adjustment and SSL necessary for understanding LT recognition.

Problem definition. LT recognition involves training a model using a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ that follows an LT distribution. In this distribution, several head classes encompass the majority of samples, whereas many tail classes have relatively few samples. The model is tasked with mapping images from input space X to classes in target space Y= $\{1, 2, \ldots, K\}$. Mapping function φ is typically composed of neural networks for backbone feature extractor $F : X \to Z \in \mathbb{R}^h$ and linear classifier $G : Z \to Y$. Generally, deep models trained on datasets with LT tendencies perform well on the head classes but exhibit significant performance degradation on the tail classes, leading to imbalanced generalisation. To address this, specialised learning strategies are required to mitigate the bias towards the head classes in the model and achieve balanced recognition across all classes.

Logit adjustment [43] is a method that modifies the loss margin based on the prior probabilities of each class. It corrects the model's bias using the prior probabilities of each class as the margin. To achieve this, given N samples, the prior probability π_y of each class y, and the corresponding class logit φ_y, the logit adjustment loss $\mathcal{L}_{LA}$ is defined as follows:

$$\mathcal{L}_{LA} = -\sum_{i=1}^{N} \log \left(\frac{e^{\varphi_y(x) + \log \pi_y}}{\sum_{y' \in Y} e^{\varphi_{y'}(x) + \log \pi_{y'}}} \right). \tag{1}$$

Through this loss function, the model reduces bias based on the class distribution and can more effectively recognise tail classes with relatively few samples.

SCL is a generalised version of contrastive learning designed to handle labelled data. It aims to maximise the similarity between samples of the same class while learning to distinguish samples from different classes. The SCL loss can be expressed as follows:

$$\mathcal{L}_{SCL} = -\frac{1}{N_B} \sum_{p \in A(y_i)} \log \frac{exp(z_i \cdot z_p/\tau)}{\sum_{y' \in Y} \sum_{k \in A(y')} exp(z_i \cdot z_k/\tau)}. \tag{2}$$

where $N_B$represents a mini-batch $B = \{x_i\}_{i=1}^{N}$ of size N , $A(y_i) = \{j \in B \mid y_j = y_i,\ j \neq i\}$ represents the set of instance indices belonging to class y_i in batch B, and $z_i = \frac{F(x_i)}{F(x_i)}$ denotes the normalised feature vector. This loss function calculates the average over the positive pairs for all samples.

Problem of SCL in LT Recognition. When SCL is applied to a dataset with an LT distribution, the problem of imbalanced feature-space formation arises. In an LT distribution, the number of samples per class varies, causing the model to learn the head-class features more effectively while struggling with tail-class feature learning. Consequently, the feature space for the head classes becomes densely populated, whereas the feature space for the tail classes remains underdeveloped because of the lack of samples. In addition, a lack of diversity within the same class reduces sample interactions and the availability of hard negative samples, making effective discrimination challenging. During training, head class samples frequently appear, whereas tail class samples appear infrequently, leading to model overfitting on the head classes and inaccurate feature representations for the tail classes. As a result, applying SCL on datasets with an LT distribution degrades the prediction performance for the tail classes, reduces the generalisation ability of the model, and creates an imbalanced feature space.

3.2 BCCL

To address the issues with SCL mentioned above, this paper proposes balanced clustering to ensure that the samples are evenly distributed in the feature space. Balanced clustering involves clustering the features extracted by a model to compress the space of the head class and expand the space of the tail class, thereby constructing a balanced feature space. To achieve this, class-specific queues are used, and clustering is performed within each queue. Unlike contrastive learning, which minimises the distance between positive samples, the cluster centroids loss increases the distance between the cluster centroids and samples to maintain balance in the feature space.

Class-wise queue-based clustering. In datasets with an LT distribution, class imbalance occurs during training because not all classes are present in each minibatch. Suh et al. [24] addressed this issue using class-wise queues. However, this approach sets different queue sizes for each class, which can still result in bias during clustering. Therefore, this study proposes a queue-based clustering method that designates a fixed-size queue for each class and performs clustering within each queue to prevent imbalance both within and across queues. The class-wise queues Q_k for the feature vectors required for clustering are constructed as follows:

$$Q_k \leftarrow \left\{ \begin{array}{ll} append(F(x_i)) & if\ |Q_k| < M \\ pop_{front(Q_k)},\ append(F(x_i)) & if\ |Q_k| \geq M \end{array} \right\}. \tag{3}$$

where $F(x_i)$ denotes the feature obtained through encoder F, and x_i represents a sample from mini-batch B. In addition, Q_k is a class-wise queue, where for each class k, $Q_k = \{F(x_i) \mid y_i = k\}$, and M is a hyperparameter that limits the size of the queue for each class.

As shown in Fig. 2, input x_i' is augmented and fed into the encoder, which then updates the class-specific queues. If the number of samples in the queue is less than M, the queue is updated directly. When the queue size reaches or exceeds

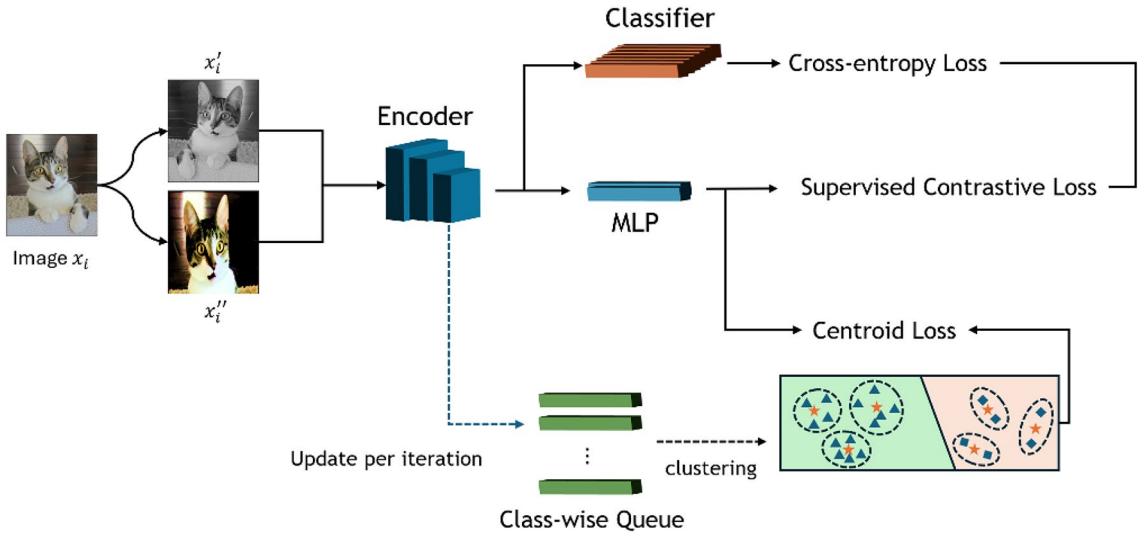

Fig. 2 Overall architecture of the proposed model. Input image x_i is first augmented and then transformed into features by the encoder. These features are fed into two learning branches and the class-specific queues. The class-specific queues are updated at each iteration, and multiple centroids are extracted through clustering. The model is trained using three different loss functions

M, the oldest samples are removed before the update. In the early stages of training, feature representations are not well formed, and because of the LT distribution, not all classes appear in a mini-batch, making clustering infeasible. Therefore, a warm-up stage is employed to enhance the feature representation and update the queues.

Class-wise centroid extraction. To construct a balanced feature space across classes, we replace each class with multiple centroids in a manner similar to that in [14]. The extraction of centroids from class-specific queues, updated during the warm-up stage, effectively reflects the features of each class by utilising clustering labels. By leveraging the constructed class-specific queues, we extract a set of several centroids $\boldsymbol{C}$ using the k-means clustering algorithm and the features $\boldsymbol{Z} = \{F(x_i)\}_{i=1}^{N}$ within the queue as follows:

$$\boldsymbol{C} = \{c_k\}_{k=1}^{k} = k - menas(\mathbf{Z}, k). \tag{4}$$

3.3 Balanced feature space construction

In this section, we describe the centroid loss for maintaining balance across classes in the feature space using the extracted centroids. We also describe the class complement, which is used to address the class imbalances in the mini-batches during training on an LT distribution.

To design a balanced feature space, two types of loss functions are utilised: SCL, which aims to adjust the distances between feature vectors, training instances of the same class to be closer together and instances of different classes to be farther apart in the feature space. Centroid loss is necessary because the feature space learned by SCL for an LT distribution tends to be biased towards the head class. Therefore, to dynamically adjust the balanced feature space, it is necessary to shrink the feature space of the head class and expand that of the tail class. As depicted in Fig. 2, the feature space is dynamically adjusted using the centroids extracted from the class-wise queues. Specifically, the centroid loss expands the feature space of tail classes to enable a more balanced and discriminative representation across the entire dataset. We construct a balanced feature space using the centroids obtained through queue-based clustering. The centroid loss functions are defined as follows:

$$\mathcal{L}_{CL} = -\frac{1}{N_B + |\boldsymbol{C}|} \sum_{p \in A(y_i^{sc}) \cup C_y^{sc}} \log \frac{exp(z_i \cdot z_p/\tau)}{\sum_{y' \in Y} \sum_{a \in A(y') \cup C'} \sum_{k \in a} exp(z_i \cdot z_k/\tau)}. \tag{5}$$

here C_y^{sc} represents the samples within the same cluster as the centroid for class y, $|\bullet|$ denotes the cardinality of the set, and y_i^{sc} denotes the set of samples for subclass y_i^{sc}. The centroid loss aims to maintain closer distances between samples and centroids of the same subclass and discriminative distances between centroids of different subclasses.

Class complement. Because of the LT distribution, not all classes may appear in all mini-batches. To address this, the class complement is performed by including the centroids obtained through the class-wise queue as samples as follows:

$$\mathcal{L}'_{SCL} = -\frac{1}{N_B + |\boldsymbol{C}|} \sum_{p \in A(y_i) \cup C_y} \log \frac{exp(z_i \cdot z_p/\tau)}{\sum_{y' \in Y} \sum_{k \in A(y') \cup C'} exp(z_i \cdot z_k/\tau)}. \tag{6}$$

Specifically, the equation above is generated by replacing y_i^{sc} with y_i and C_y^{sc} with C_y in Eq. (5). The class complement adjusts the imbalance in mini-batches by incorporating all classes into the loss function calculation during training, thereby mitigating the learning imbalance caused by the LT distribution.

Logit compensation. In tasks involving LT distributions, data imbalance can cause bias in the logits generated by the final classification layer. The purpose of logit compensation is to address this bias by adjusting the decision boundary to account for data imbalance. This method can be applied during both the training and test phases. Previous studies [44–46] have demonstrated the effectiveness of logit compensation in LT recognition, and its general form can be expressed as follows:

$$\lambda_i = \log\left(\frac{n_i}{\sum_{j=1}^{Y} n_j}\right), \quad i = 1, 2, \ldots, Y. \tag{7}$$

where λ_i represents the class-wise log probabilities used as weights for the data distribution in logit compensation. In the logit compensation loss, which is calculated as follows:

$$\mathcal{L}_{LC} = -\sum_{i=1}^{N} \log\left(\frac{e^{(\varphi_y(x)+\lambda_i)}}{\sum_{y' \in Y} e^{(\varphi_{y'}(x)+\lambda_i)}}\right). \tag{8}$$

the input logit φ_y is adjusted to mitigate the classifier bias in LT distributions. Finally, the overall training loss is given below. During the initial training phase, the loss $\mathcal{L}_{SCL}$ from Eq. (2) is used. After the warm-up stage, the class-complement adjusted loss $\mathcal{L}'_{SCL}$ from Eq. (6) is applied. This training loss is represented as follows:

$$\begin{cases} \mathcal{L}_{total} = \mathcal{L}_{SCL} + \mathcal{L}_{CL} + \gamma\mathcal{L}_{LC}, & \text{Before warm} - \text{up stage} \\ \mathcal{L}_{total} = \mathcal{L}'_{SCL} + \mathcal{L}_{CL} + \gamma\mathcal{L}_{LC}, & \text{After warm} - \text{up stage} \end{cases}. \tag{9}$$

where $\mathcal{L}_{SCL}$ and $\mathcal{L}_{CL}$ have the same weight, but $\mathcal{L}_{LC}$ is influenced by the hyperparameter γ because it addresses bias and adjusts the decision boundary. The value of γ is determined experimentally, and a detailed description of this process is provided in Sect. 4.3.

4 Experiment

LT visual recognition is characterised by an imbalanced distribution of the data, where a few head classes occupy the majority of the samples in the dataset, while many tail classes occupy a small fraction of the entire dataset. In our experiments, we used the following four representative datasets for image classification under LT distribution: CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018.

CIFAR-10-LT/CIFAR-100-LT are versions of the original CIFAR-10 and CIFAR-100 datasets modified so that they have LT distributions. These datasets were created by sampling the original training sets of CIFAR-10 and CIFAR-100, which consist of 10 and 100 classes, respectively. CIFAR-10-LT and CIFAR-100-LT were constructed by adjusting the number of samples in each class of the training set. For instance, the imbalance factor, which indicates the degree of imbalance, is defined as the number of samples in the class with the most samples divided by the number of samples in the class with the fewest samples. In the experiments, imbalance factor values of 10, 50, and 100 were used to generate various LT training sets. Balanced validation sets from the original CIFAR-10 and CIFAR-100 datasets were used for testing.

ImageNet-LT is a modified version of the standard ImageNet dataset, created by selecting images following an indicator of Pareto distribution $\alpha_p = 6$. α_p is a parameter of the Pareto distribution that determines the degree of imbalance in the dataset. A larger value of α_p results in a more skewed distribution, where a few categories dominate in terms of the number of samples (head classes), while the majority of categories have very few samples (tail classes). For ImageNet-LT, $\alpha_p = 6$ ensures a severe LT distribution suitable for evaluating LT recognition methods. This dataset consists of 115,800 images across 1000 categories, with the number of images per category ranging from 1280 to 5.

iNaturalist 2018 is a large-scale dataset consisting of 437,500 images across 8,142 classes, and exhibits a highly imbalanced distribution. The number of images per class ranges from 2 to 1000, indicating a significant degree of imbalance.

To systematically evaluate the performance under various levels of data availability, we followed the categorisation method proposed in [47]. The categories in these datasets were divided into three subsets based on the number of training samples per category. The many-shot categories contained more than 100 training images, medium-shot categories contained between 20 and 100 training samples, and few-shot categories contained fewer than 20 training images.

4.1 Experiment setup

Backbone networks. For experiments on CIFAR-10/100-LT, we utilised ResNet-32, and to ensure a fair comparison with GLMC, we modified the backbone using the same settings. For ImageNet-LT and iNaturalist 2018, we employed ResNet-50 and ResNeXt-50-32×4d as backbone models,

respectively. We adhered to the training settings in [13] to ensure a fair evaluation.

we perform k-means clustering every 5 epochs. This frequency was chosen based on empirical observations to balance computational efficiency and clustering accuracy. Performing clustering less frequently (e.g., every epoch) introduced excessive computational overhead without noticeable improvements in feature distribution. Conversely, performing clustering too infrequently (e.g., every 10 epochs) delayed convergence, particularly in balancing tail class representations. Thus, the 5-epoch interval provides an optimal trade-off, ensuring that cluster centroids are regularly updated while maintaining efficient training.

CIFAR-10/100-LT. Similar to prior studies [50, 51], AutoAug [52] and Cutout [53] were employed as data augmentation techniques for the classification head, and SimAug [31] was used for contrastive learning. A batch size of 128 and weight decay of 5e-4 were used. The model was trained for 200 epochs with a learning rate of 0.15 and a cosine scheduler. The warm-up epochs was conducted for the first 20 epochs to stabilize feature representations before incorporating the centroid loss. For contrastive learning, the temperature parameter τ and projection head for representation learning have an output dimension of 128 and hidden layer dimension of 512.

ImageNet-LT and iNaturalist 2018. For the classification head, we used RandAug [54], and for contrastive learning, SimAug [31] was employed. The model was trained for 100 epochs with a batch size of 128 and an initial learning rate of 0.1 using a cosine scheduler. The warm-up epochs was conducted for the first 10 epochs to stabilize feature representations before incorporating the centroid loss. The representation part included a projection head with an output dimension of 2,048 and a hidden layer dimension of 1,024, and the temperature parameter τ for contrastive learning was set to 0.07. The weight decay for ImageNet-LT was set to 5e-4 and that for iNaturalist 2018 was set to 1e-4.

The models were trained on CIFAR-10/100-LT on a single Nvidia GeForce RTX 2080Ti GPU, whereas those trained on ImageNet-LT and iNaturalist 2018 datasets were trained on a single Nvidia GeForce 3090 GPU.

4.2 Benchmark results

To evaluate the performance of the proposed BCCL, experiments were conducted using CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018 datasets.

Results on CIFAR-10/100-LT. As the results in Table 1 reveal, BCCL consistently demonstrated competitive performance across various imbalance factors when compared with existing state-of-the-art (SoTA) methods. This indicates that BCCL can learn more effectively than conventional methods by constructing a balanced feature space for each class. Notably, BCCL improved the performance of the tail classes while maintaining the performance of the head classes. This outcome addresses the significant bias problem in conventional contrastive learning methods, demonstrating that BCCL can simultaneously maintain interclass and intraclass balances through multi-clustering. Furthermore, an evaluation of the generalisation ability of BCCL in various environments revealed that BCCL achieved SoTA performance across all experimental datasets. It is particularly noteworthy that BCCL outperformed GLMC even with the number of channels in the backbone model increased fourfold. These results suggest that BCCL effectively mitigates the bias of deep models in LT-distributed datasets and maintains a balanced feature space. Table2 presents the performance on the CIFAR-100-LT dataset across different shots with varying sample sizes. The proposed BCCL method outperformed all other methods, demonstrating

Table 1 Performance comparison on CIFAR-100-LT and CIFAR-10-LT

Dataset	CIFAR-100-LT			CIFAR-10-LT		
Imbalance Factor	100	50	10	100	50	10
Logit Adjustment [43]	50.5	54.9	64.0	84.3	87.1	90.9
KCL [48]	42.8	46,3	57.6	77.6	81.7	88.0
TSC [49]	43.8	47.4	59.0	79.7	82.9	88.7
BCL [13]	51.93	56.69	64.87	84.32	87.24	91.12
SBCL [14]	44.9	48.7	57.9	–	–	–
CUDA [38]	52.3	56.2	64.6	–	–	–
GLMC	43.55	52.69	62.27	84.47	86.53	90.92
BCCL	**52.55**	**56.89**	**65.3**	**85.22**	**88.07**	**91.92**
GLMC [25]*	55.88	61.08	70.74	87.75	90.18	94.04
BCCL*	**57.36**	**62.66**	**73.01**	**87.84**	**90.48**	**94.94**

Top-1 accuracy of ResNet-32

The best results are marked in bold

* indicates that the channel size of the ResNet-32 backbone was increased by a factor of four following the settings of GLMC [23]

Results were obtained at 200 epochs for each dataset

Table 2 Performance comparison for each shot on CIFAR-100-LT

Methods	Many	Medium	Few	All
Logit Adjustment [43]	67.2	51.9	29.5	50.5
KCL [48]	63.4	42.5	19.2	42.8
DRO-LT [55]	64.7	50.0	23.8	47.3
BCL [13]	67.2	53.1	32.9	51.93
SBCL [14]	64.4	45.3	22.2	44.9
BCCL	68.1	53.9	34.5	52.55

Top-1 accuracy is reported on ResNet-32

'Many', 'Medium', and 'Few' indicate classes with varying numbers of samples, while 'All' indicates the performance on the entire test dataset

Table 3 Performance comparison on ImageNet-LT and iNaturalist

Dataset	ImageNet-LT	iNaturalist 2018
MetaSAug [41]	47.4	65.6
SCL [12]	49.8	66.4
LWS [47]	49.9	65.9
KCL [48]	51.5	68.6
DisAlign [55]	52.9	69.5
SBCL [14]	53.4	70.8
GCL [56]	54.9	72.0
BCL [13]	56.0	71.8
BCCL	57.1	72.3

Top-1 accuracy on ResNet-50 is reported

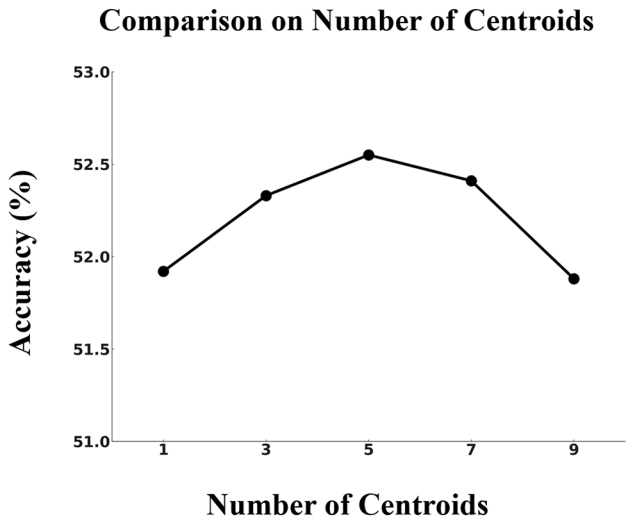

Fig. 3 Performance comparison on CIFAR-100-LT with an imbalance factor of 100 based on the number of centroids

its effectiveness in improving the performance for each class in imbalanced d atasets. These results indicate that BCCL effectively addresses class imbalance, leading to an improvement in overall performance

Results on ImageNet-LT and iNaturalist 2018. Table 3 presents the overall Top-1 accuracy results for ImageNet-LT and iNaturalist 2018. BCCL outperformed existing models, demonstrating the need for well-represented learning in large-scale datasets to achieve performance improvements.

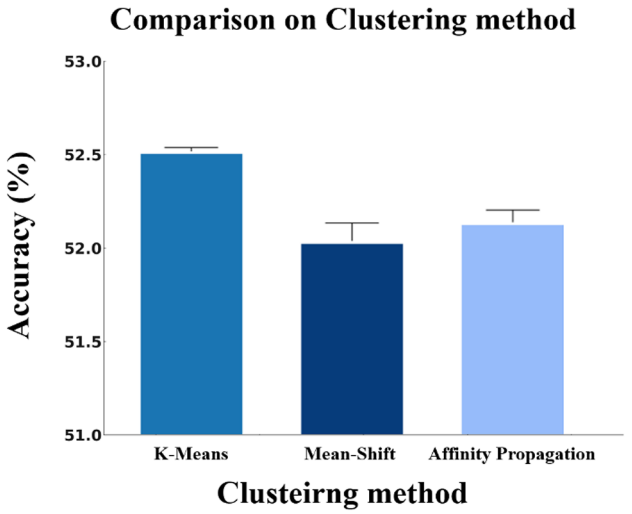

Fig. 4 Performance comparison on CIFAR-100-LT with an imbalance factor of 100 according to clustering method. Different colors indicate different clustering methods

In addition, the use of clustering in BCCL proved effective in constructing a balanced feature space, not only between classes but also within classes.

4.3 Components evaluation

Number of centroids for a balanced feature space. An ablation study was conducted on the number of centroids required for class-complement and intraclass balance. Figure 3 quantitatively evaluates the impact of the number of centroids on model performance. The experimental results indicate that setting the number of centroids to five achieves the highest accuracy (52.55%), suggesting that five centroids optimally balances the interclass and intraclass features. Conversely, having only one or nine centroids resulted in the lowest accuracy, highlighting the limitations of a single centroid and the potential performance degradation due to excessive centroids. These findings underscore the importance of appropriately setting the number of centroids to effectively capture the characteristics of the data and enhance the generalisation performance of the model.

Clustering method evaluation. Figure 4 shows an analysis of the impact of three clustering methods—K-means, mean-shift, and affinity propagation–on model performance. Despite the drawback of needing to predefine the number of centroids, K-means was shown to be fast and efficient, effectively leveraging the advantages of centroids. In contrast, mean-shift exhibited a lower performance with an accuracy of 52.0%, because of its high computational cost and time consumption. Affinity propagation, achieved an accuracy of 52.2% and maintained reasonable performance as a similarity-based clustering method. These results suggest that centroid-based models can maximise performance

through efficient clustering methods without being significantly affected by the choice of clustering algorithm.

Hyperparameter γ for centroid loss. Figure 5 analyses the impact of hyperparameter γ in the centroid loss within the total loss on the model's performance. As observed in the results, the highest accuracy of 52.55% is achieved when γ is set to 0.1. This demonstrates the ability to appropriately adjust γ to enhance the model's generalisation capacity without disrupting the existing SCL. Particularly during the training for the tail classes, to expand the feature space to achieve balance, γ was fixed at 0.1. The accuracy consistently decreases as γ increases. These findings underscore the significant influence of the hyperparameter settings in the centroid loss on model performance, emphasising the importance of selecting an appropriate γ value.

4.4 Ablation study

Supervised Contrastive Learning. The supervised contrastive loss ($\mathcal{L}_{SCL}$) serves as the foundation for feature learning in the proposed method. It encourages feature compactness within the same class and feature separability across different classes, providing a strong baseline for LT recognition. However, as evident from the Table 4, when applied alone, $\mathcal{L}_{SCL}$ underperforms due to its inability to handle the inherent class imbalance in LT datasets. The head classes dominate the feature space, leaving the tail classes poorly represented.

Centroid Loss. The centroid loss ($\mathcal{L}_{CL}$) addresses the imbalance in the feature space by compressing the feature representations of head classes and expanding those of tail classes. When combined with $\mathcal{L}_{SCL}$, it significantly enhances the model's ability to learn balanced feature distributions, resulting in improved performance across all imbalance factors. This improvement highlights the importance of explicitly designing the feature space to mitigate class imbalance.

Logit Compensation Loss. The logit compensation loss ($\mathcal{L}_{LC}$) complements $\mathcal{L}_{SCL}$ by adjusting the classifier's decision boundaries based on the class distribution. By introducing this component, the model compensates for the bias towards head classes, improving the recognition of underrepresented tail classes. Although $\mathcal{L}_{LC}$ alone does not address the feature space imbalance, its synergy with other losses contributes to more robust and unbiased decision-making.

Total Loss. The combination of $\mathcal{L}_{total}$ achieves the best performance across all imbalance factors. The results demonstrate that $\mathcal{L}_{SCL}$ provides a solid foundation for feature learning, while $\mathcal{L}_{CL}$ and $\mathcal{L}_{LC}$ play complementary roles in balancing the feature space and mitigating classifier bias, respectively. This synergy ensures that both the feature space and classifier outputs are optimized for LT distributions, resulting in state-of-the-art performance.

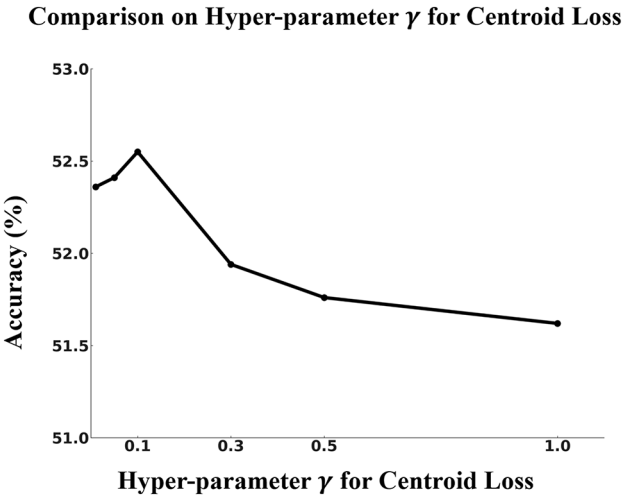

Fig. 5 Performance comparison on CIFAR-100-LT with an imbalance factor of 100 for different values of γ in the centroid loss, which affects BCCL training

Table 4 ù on CIFAR-100-LT

$\mathcal{L}_{SCL}$	$\left(\mathcal{L}'_{SCL}\right)$	$\mathcal{L}_{LC}$	CIFAR-100-LT		
Imbalance Factor			100	50	10
√			48.37	52.18	59.21
√	√		50.57	54.83	64.46
√		√	50.44	55.31	63.87
√	√	√	52.55	56.89	65.3

5 Conclusion

In this study, we proposed BCCL to address the visual recognition problem under LT distributions. BCCL aims to align representations between head and tail classes more effectively through queue-based multi-clustering and adaptive feature space adjustment. Specifically, by utilising class-wise queues to extract multiple centroids and dynamically adjust the feature space of each class, we mitigated the imbalance inherent in LT distributions. We evaluated the performance of BCCL on various LT benchmark datasets such as CIFAR10-LT, CIFAR100-LT, ImageNet-LT, and iNaturalist 2018. The experimental results consistently demonstrated that BCCL outperforms existing SoTA methods by maintaining a balance among classes, resolving the omission of classes within mini-batches, and expanding the feature space of the tail classes. Based on these findings, we provide evidence of the efficacy of BCCL in forming more appropriate feature spaces under LT distributions than conventional LT learning methods.

Author contributions B.I.K was responsible for the design and overall investigation. B.C.K. was responsible for the data curation, supervision, writing and editing of manuscript.

Funding This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education [2022R1I1A3058128].

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
2. 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
3. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
4. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell
5. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
6. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255
7. Wu TY, Morgado P, Wang P, Ho CH, Vasconcelos N (2020) Solving long-tailed recognition with deep realistic taxonomic classifier. In: Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, Part VIII 16. Springer International Publishing, pp 171–189
8. 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
9. Bai J, Liu Z, Wang H, Hao J, Feng Y, Chu H, Hu H (2023) On the effectiveness of out-of-distribution data in self-supervised long-tail learning. arXiv:2306.04934
10. Dong B, Zhou P, Yan S, Zuo W (2022) LPT: long-tailed prompt tuning for image classification. In: The eleventh international conference on learning representations
11. Liu B, Li H, Kang H, Hua G, Vasconcelos N (2022) Breadcrumbs: adversarial class-balanced sampling for long-tailed recognition. In: European conference on computer vision. Cham: Springer Nature Switzerland, pp 637–653
12. Khosla P, Teterwak P, Wang C, Sarna A, Tian Y, Isola P, Maschinot A, Liu C, Krishnan D (2020) Supervised contrastive learning. Adv Neural Inf Process Syst 33:18661–18673
13. Zhu J, Wang Z, Chen J, Chen YP, P, Jiang YG (2022) Balanced contrastive learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6908–6917
14. Hou C, Zhang J, Wang H, Zhou T (2023) Subclass-balancing contrastive learning for long-tailed recognition. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5395–5407
15. Li J, Meng Z, Shi D, Song R, Diao X, Wang J, Xu H (2023) Fcc: feature clusters compression for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 24080–24089
16. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images
17. 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
18. Van Horn G, Mac Aodha O, Song Y, Cui Y, Sun C, Shepard A, Adam H, Perona P, Belongie S (2018) The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8769–8778
19. Yu S, Guo J, Zhang R, Fan Y, Wang Z, Cheng X (2022) A re-balancing strategy for class-imbalanced classification based on instance difficulty. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 70–79
20. 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
21. Hong Y, Han S, Choi K, Seo S, Kim B, Chang B (2021) Disentangling label distribution for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6626–6636
22. Kim H, Kim S, Ahn D, Lee JT, Ko BC (2024) Scene graph generation strategy with co-occurrence knowledge and learnable term frequency. In: International conference on machine learning
23. Wang H, Yao M, Xu Y, Liu H, Jia W, Fu X, Wang Y (2024) Manifold-based Incomplete multi-view clustering via bi-consistency guidance. IEEE Trans Multimed 26:10001–10014
24. Suh MK, Seo SW (2023) Long-tailed recognition by mutual information maximization between latent features and ground-truth labels. In: International conference on machine learning, pp 32770–32782
25. Du F, Yang P, Jia Q, Nan F, Chen X, Yang Y (2023) Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 15814–15823
26. Wang Y, Zhang P, Bai L, Xue J (2023) Fend: a future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1400–1409
27. Du Y, Shen J, Zhen X, Snoek CG (2023) Superdisco: super-class discovery improves visual recognition for the long-tail. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 19944–19954
28. Parisot S, Esperanca PM, McDonagh S, Madarasz TJ, Yang Y, Li Z (2022) Long-tail recognition via compositional knowledge transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6939–6948
29. Kim BI, Ko BC (2024) Active contrastive learning with noisy labels in fine-grained classification. In: 2024 International conference on electronics, information, and communication (ICEIC). IEEE, pp 1–5
30. Wang Y, Peng J, Wang H, Wang M (2022) Progressive learning with multi-scale attention network for cross-domain vehicle re-identification. Sci China Inf Sci 65:6:160173

31. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: International conference on machine learning, pp 1597–1607
32. He K, Fan H, Wu Y, Xie S, Girshick R (2020) Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9729–9738
33. Cui J, Zhong Z, Liu S, Yu B, Jia J (2021) Parametric contrastive learning. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 715–724
34. Li B (2022) Adaptive hierarchical representation learning for long-tailed object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2313–2322
35. Guan Q, Li Z, Zhang J, Huang Y, Zhao Y (2023) Joint representation and classifier learning for long-tailed image classification. Image Vis Comput 137:104759
36. Ma Y, Jiao L, Liu F, Yang S, Liu X, Chen P (2024) Geometric prior guided feature representation learning for long-tailed classification. Int J Comput Vis. https://doi.org/10.1007/s11263-024-01983-2
37. Du C, Wang Y, Song S, Huang G (2024) Probabilistic contrastive learning for long-tailed visual recognition, IEEE Trans Pattern Anal Mach Intell
38. Ahn S, Ko J, Yun SY (2023) Cuda: curriculum of data augmentation for long-tailed recognition, arXiv:2302.05499
39. Hong Y, Zhang J, Sun Z, Yan K (2022) Safa: Sample-adaptive feature augmentation for long-tailed image classification. In: European conference on computer vision. Cham: Springer Nature Switzerland, pp 587–603
40. Xu Z, Chai Z, Yuan C (2021) Towards calibrated model for long-tailed visual recognition from prior perspective. Adv Neural Inf Process Syst 34:7139–7152
41. Li S, Gong K, Liu CH, Wang Y, Qiao F, Cheng X (2021) Metasaug: meta semantic augmentation for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5212–5221
42. Chu P, Bian X, Liu S, Ling H (2020) Feature space augmentation for long-tailed data. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16. Springer International Publishing, pp 694–710
43. Menon AK, Jayasumana S, Rawat AS, Jain H, Veit A, Kumar S (2020) Long-tail learning via logit adjustment, arXiv:2007.07314
44. Wu T, Liu Z, Huang Q, Wang Y, Lin D (2021) Adversarial robustness under long-tailed distribution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8659–8668
45. Kini GR, Paraskevas O, Oymak S, Thrampoulidis C (2021) Label-imbalanced and group-sensitive classification under overparameterization. Adv Neural Inf Process Syst 34:18970–18983
46. Tang K, Huang J, Zhang H (2020) Long-tailed classification by keeping the good and removing the bad momentum causal effect. Adv Neural Inf Process Syst 33:1513–1524
47. Kang B, Xie S, Rohrbach M, Yan Z, Gordo A, Feng J, Kalantidis Y (2019) Decoupling representation and classifier for long-tailed recognition, arXiv:1910.09217
48. Kang B, Li Y, Xie S, Yuan Z, Feng J (2020) Exploring balanced feature spaces for representation learning. In: International conference on learning representations
49. Li T, Cao P, Yuan Y, Fan L, Yang Y, Feris R, 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
50. Cui Y, Jia M, Lin TY, Song Y, Belongie S (2019) Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9268–9277
51. Cao K, Wei C, Gaidon A, Arechiga N, Ma T (2019) Learning imbalanced datasets with label-distribution-aware margin loss. Adv Condens Matter Phys. https://doi.org/10.1007/s11263-024-01983-2
52. Cubuk ED, Zoph B, Mane D, Vasudevan V, Le QV (2019) Autoaugment: learning augmentation strategies from data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 113–123
53. DeVries T, Taylor GW (2017) Improved regularization of convolutional neural networks with cutout, arXiv:1708.04552
54. Cubuk ED, Zoph B, Shlens J, Le QV (2020) Randaugment: practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp 702–703
55. Samuel D, Chechik G (2021) Distributional robustness loss for long-tail learning. In Proceedings of the IEEE/CVF international conference on computer vision, pp 9495–9504
56. Li M, Cheung YM, Lu Y (2022) Long-tailed visual recognition via gaussian clouded logit adjustment. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6929–6938