Pattern Recognition 147 (2024) 110107

Contents lists available at ScienceDirect

Pattern Recognition

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

DBN-Mix: Training dual branch network using bilateral mixup augmentation for long-tailed visual recognition☆

Jae Soon Baik, In Young Yoon, Jun Won Choi *

Department of Electrical Engineering, Hanyang University, Republic of Korea

ARTICLE INFO

Keywords:
Long-tailed visual recognition
Class imbalance
Image classification
Mixup augmentation
Temperature scaling

ABSTRACT

There is growing interest in the challenging visual perception task of learning from long-tailed class distributions. The extreme class imbalance in the training dataset biases the model to prefer recognizing majority class data over minority class data. Furthermore, the lack of diversity in minority class samples makes it difficult to find a good representation. In this paper, we propose an effective data augmentation method, referred to as *bilateral mixup augmentation*, which can improve the performance of long-tailed visual recognition. The bilateral mixup augmentation combines two samples generated by a uniform sampler and a re-balanced sampler and augments the training dataset to enhance the representation learning for minority classes. We also reduce the classifier bias using class-wise temperature scaling, which scales the logits differently per class in the training phase. We apply both ideas to the *dual-branch network (DBN)* framework, presenting a new model, named *dual-branch network with bilateral mixup (DBN-Mix)*. Experiments on popular long-tailed visual recognition datasets show that DBN-Mix improves performance significantly over baseline and that the proposed method achieves state-of-the-art performance in some categories of benchmarks.

1. Introduction

Deep neural networks (DNNs) have achieved great success in a variety of visual perception tasks thanks to publicly available large datasets such as ImageNet [1] and MS COCO [2]. Although the classes of images in these recognition datasets are balanced to have an approximately uniform distribution, large-scale real-world datasets have a long-tailed distribution; few classes occupy most of the data, while most classes have few samples. Standard supervised learning on a long-tailed dataset tends to be severely biased toward majority classes, resulting in poor classification accuracy for minority classes. This trend is problematic in applications such as autonomous driving, where image recognition of all other classes is equally important. They raise the challenge of designing effective training methods for *long-tailed datasets*, which can improve the recognition performance for both majority and minority classes.

Various training methods for long-tailed recognition tasks have been proposed to date [3–6]. Multiple expert networks have been used to address class imbalance issues, where multiple models are jointly trained to model both majority and minority class data [5,7–11]. RIDE [5] proposed the routing method for determining a way to allocate the training samples to multiple experts. BBN [8] employed a dual-branch network (DBN) consisting of two parallel branches called a *conventional learning branch* and a *re-balancing branch*. The conventional learning branch was trained using a *uniform sampler* while the re-balancing branch was trained using a *re-balanced sampler*, which generated the samples with inversely proportional class distribution to the original dataset. The re-balancing branch can effectively alleviate the classifier bias, but it tends to oversample minority class samples, thereby degrading the quality of the representation.

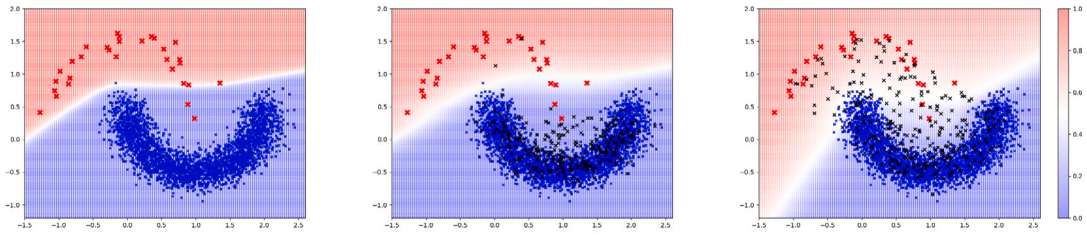
An alternative method for addressing the class imbalance issue is data augmentation [12–20]. This method is particularly useful for long-tailed datasets, as synthetically generated samples can mitigate the lack of minority class data. Various data augmentation strategies have been proposed for long-tailed recognition task [12,13,16–20]. DFG [13] integrated a feature extractor and the widely used GAN structure [21] with attention models to generate discriminative features for minority classes. MetaSAug [15] adopted a meta-learning method to augment diverse samples in semantically meaningful ways. However, due to its limited size and diversity, it is difficult to generate samples that follow the true distribution of minority class data.

In this paper, we present a simple yet effective data augmentation strategy, referred to as *bilateral mixup augmentation*, designed to address

☆ This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2021-0-01314, Development of driving environment data stitching technology to provide data on shaded areas for autonomous vehicles).

* Corresponding author.
E-mail addresses: jsbaik@spa.hanyang.ac.kr (J.S. Baik), inyoungyoon@spa.hanyang.ac.kr (I.Y. Yoon), junwchoi@hanyang.ac.kr (J.W. Choi).

https://doi.org/10.1016/j.patcog.2023.110107
Received 17 November 2022; Received in revised form 31 October 2023; Accepted 1 November 2023
Available online 10 November 2023



(a) Empirical risk minimization (b) Mixup (c) Bilateral Mixup

Fig. 1. Outputs of the classifiers trained by (a) empirical risk minimization (ERM), (b) mixup, and (c) bilateral mixup: Three-layer neural networks were trained on the imbalanced two half-moon dataset with the imbalance ratio of 100. Red and blue cross marks correspond to minority class samples and majority class samples, respectively. The black cross marks indicate the training samples generated by the data augmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the class imbalance for long-tailed recognition tasks. The proposed bilateral mixup differs from conventional mixup [22] in that a *class distribution-aware mixup strategy* is used to combine the samples from a uniform sampler and a re-balanced sampler. The samples generated by the proposed mixup operations are located near the boundaries of minority class regions, where data points are sparsely distributed, and serve to better capture the distribution of minority classes as distinct from other classes. Our class distribution-aware combination rule significantly enhances the ability of the original mixup augmentation to improve data representation, especially for long-tailed class distributions. Fig. 1 shows a toy example that demonstrates the effect of the proposed bilateral mixup as compared to the original mixup. Without the proposed combination rule, minority class samples would not participate in sample generation, failing to generate a sufficient number of augmented minority samples (see Fig. 1(b)). As a result, the conventional mixup augmentation does not produce a decision boundary that well separates two different classes. Note that the convention mixup barely improves the decision boundary obtained by the empirical risk minimization (ERM) in Fig. 1(a). On the other hand, Fig. 1(c) shows that the proposed bilateral mixup generates augmented samples in the region between majority and minority classes, thereby producing a decision boundary that better separates the minority class samples from the other samples.

While bilateral mixup improves the representation ability, we also need a measure to compensate for the bias of the classifier. We present a *class-wise temperature scaling* method that applies class-dependent temperature parameters to the logits of the classifier. The proposed class-wise temperature scaling can be readily applied to the standard cross-entropy loss. Our extensive experiments show that the proposed temperature scaling complements the bilateral mixup augmentation by mitigating the bias of the classifier.

We integrate the above two ideas into the multiple expert network framework, presenting a new architecture, the so-called *dual-branch network with bilateral mixup (DBN-Mix)*. We extensively evaluated the performance of the proposed DBN-Mix on widely used long-tailed visual recognition datasets: CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018. The proposed DBN-Mix significantly outperforms conventional training methods designed for long-tailed visual recognition and achieves state-of-the-art performance in some categories of benchmarks.

The main contributions of this study are summarized as follows

- We present a simple yet effective data augmentation method designed to improve long-tailed visual recognition performance. We propose a novel mixup operation that combines two samples drawn from the dataset with different sampling distributions. The class distribution-aware mixup strategy serves to better model minority classes, improving classification accuracy.
- Our study addresses two sources of performance degradation caused by long-tailed class distributions: (1) poor representation of minority class data due to lack of diversity in samples and (2) classifier bias caused by imbalanced class distribution. Our analyses show that the proposed bilateral mixup and class-wise temperature scaling effectively mitigate the performance degradation in both representation learning and classifier learning. While numerous existing methods have used two-stage training to improve both representation learning and classifier learning [18, 23–26], our class-dependent mixup method enables *end-to-end training*, which results in improved performance.
- The proposed method can be implemented only with a few lines in the code and does not require a complex optimization process like other data augmentation methods [17,18]. Furthermore, since the bilateral mixup operation only affects the input samples without changing the main network, it can be easily extended to any network architecture for long-tail visual recognition tasks. We demonstrate that our ideas can also be simply integrated into the common single-branch network (SBN) and achieve significant performance improvements over the baseline, although its classification accuracy is not as high as that achieved by DBN-Mix.

2. Related work

2.1. Re-sampling and re-weighting

Re-sampling and re-weighting methods have been extensively researched. The re-sampling strategy balances the class distribution by oversampling minority class data or undersampling majority class data. Conventional oversampling methods [14,27–31] to address class imbalance include SMOTE [14], Borderline-SMOTE [27], Random oversampling [32], and Relay BP [29]. Subsequent works, including SMOM [31], SMOTE-SF [33], FW-SMOTE [34], and SMOTE-IPF [35], have further refined and expanded the SMOTE approach. Existing undersampling methods [32,36–39] include One-sided selection [37], Random undersampling [32], C4.5 [36], and RBU [39]. A comprehensive introduction and analysis of these oversampling and undersampling strategies were presented in [32]. Re-weighting approaches modify the loss function based on class- or sample-level criteria [3,4,40–44]. Cao *et al*. [3] proposed a label-distribution-aware margin loss based on the theoretical margin bound, and Cui *et al*. [4] introduced the notion of an effective number to determine re-weighting factors. Several studies have employed meta-learning to determine a strategy for weighting the loss function [41,42,44,45]. ORESCNN [44] employed the contrastive loss from Siamese neural networks, along with a meta-learning objective, to calculate an online weight for each sample. Recently, GOL [43] mitigated the location imbalance in both long-tail instance segmentation and object detection tasks by replacing the Softmax function by Gumbel activation. In contrast, our method focuses on addressing the imbalance issue affected by class label distribution rather than location imbalance.

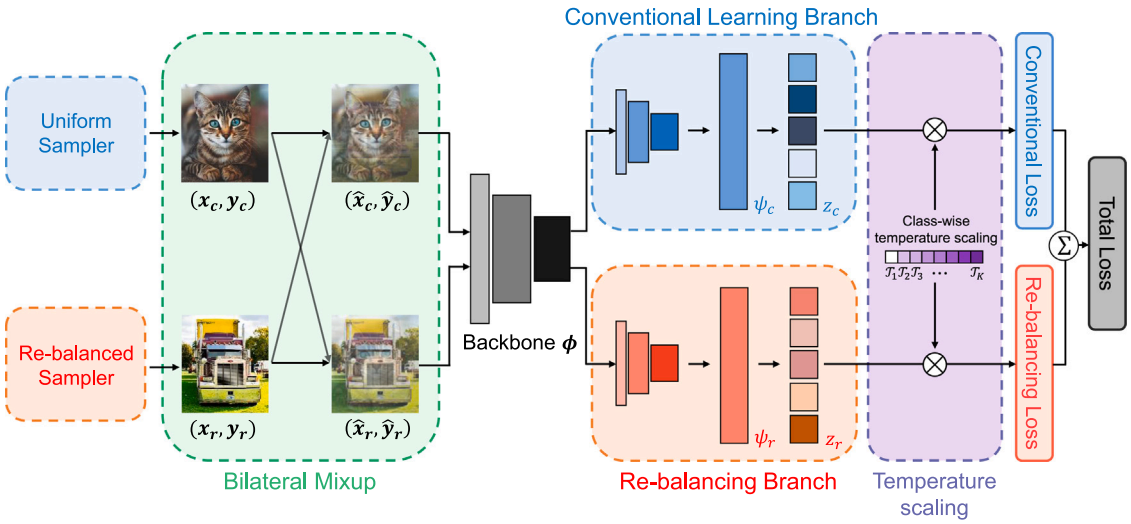


Fig. 2. Overview of the proposed method: Our method use two bilateral mixup samples $(\hat{x}_c, \hat{y}_c)$ and $(\hat{x}_r, \hat{y}_r)$ to train two branch networks, conventional learning branch and re-balancing branch. In the training phase, we use $(\hat{x}_c, \hat{y}_c)$ for conventional learning branch and $(\hat{x}_r, \hat{y}_r)$ for re-balancing branch. For the inference phase, two prediction logits from each branch are averaged to return the final output.

2.2. Two-stage training strategy

Several recent studies have investigated the impact of long-tailed class distribution on representation learning and classifier learning and have proposed two-stage training methods to improve both [6,18,23–26]. The pioneering work in [23] first used normal training data to train the backbone network and then used class-balanced samples to refine the classifier only. Since then, several two-stage training methods have been proposed, including the CAM-based method [18], logit adjustment loss [46], MetaSAug [15], MiSLAS [6], DLSA [47], and Breadcrumb [48]. DLSA [47] adopted a Gaussian mixture flow filter to build a new label space for classifier learning to increase the discrimination between majority and minority classes. Breadcrumb [48] proposed a novel adversarial class-balanced sampling strategy that uses feature traces obtained during training as augmented features. Although our method attempts to solve the class imbalance issue from both representation learning and classifier learning perspectives, the proposed DBN-Mix does not require two-stage training and allows end-to-end learning.

2.3. Ensemble-based approach

Ensemble-based methods for long-tailed visual recognition have been actively studied [5,7–11]. These methods use multiple ensemble models to model data with different class distributions. Jan et al. [9] proposed an ensemble of deep networks trained on a subspace that contains balanced and strong class-associated data clusters. BBN [8] employed two branch networks, where the conventional learning branch was trained by the uniform sampler and the re-balancing branch was trained by the re-balanced sampler. LFME [7] trained multiple networks on subsets of the dataset and then aggregated the information from the subnetworks using knowledge distillation. RIDE [5] employed the multiple networks to reduce both the model bias and variance using distribution-aware loss. In [49], the cross-branch consistency loss between two branch networks was used as a regularizer for multi-label visual recognition learning.

2.4. Data augmentation

A data augmentation method has been used to handle the data scarcity of minority classes and mitigate undesirable bias [12–20,22,50]. DFG [13] introduced a feature generation method that utilizes the supervised attention mechanism to identify and select discriminative features of the minority class data. LCReg [12] proposed feature-based augmentation that learns class-agnostic latent features common to all classes. These features are used to rectify the original features distorted by imbalance bias. As mixup augmentation has been proposed to improve the generalization of the model [22], it has been adapted to solve the problem of long-tailed visual perception [6,18,19]. Remix [19] employed the mixup strategy that assigned higher mixing factors to labels associated with minority class samples. MiSLAS [6] used the mixup only for the first stage of training and then used label-aware smoothing to deal with classifier bias. CMO [20] proposed a data augmentation method based on CutMix that can transfer rich contexts from majority to minority samples. RISDA [51] proposed implicit data augmentation [52], which uses semantic similarity information obtained from covariance matrices and knowledge graphs for data augmentation. GLMC [53] employed a cumulative learning approach for re-weighting the loss and a self-supervised consistency loss between global and local features to improve the robustness of the representation of the minority classes. Several methods [15,17,50] attempted to use the knowledge transfer to generate synthetic minority class samples using the learned representation of majority classes. SAFA [50] proposed an augmentation method to enrich the features of minority classes by utilizing class-independent information from various majority classes. However, these methods require a carefully designed optimization process to achieve their goals.

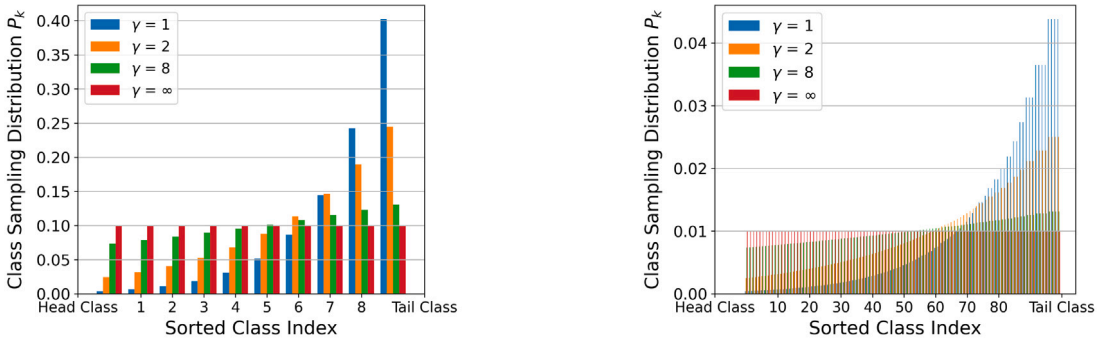
3. Proposed method

In this section, we present the details of the proposed DBN-Mix method.

3.1. Overview of DBN-Mix

Consider a K-class image classification task. We train a backbone network ϕ and a classifier network ψ in an end-to-end fashion on the long-tailed training set. Let $D = (X, Y)$ be the training dataset, where $X = \{x_1, \ldots, x_N\}$, $Y = \{y_1, \ldots, y_N\}$ and x_i and y_i are the ith image sample and the corresponding label, respectively. N denotes the cardinality of the training dataset. The label y_i is encoded by a one-hot vector $[y_{i,1}, \ldots, y_{i,K}]^T \in \{0,1\}^K$.

Fig. 2 depicts the structure of DBN-Mix. It consists of the shared backbone network ϕ followed by two branch subnetworks, the conventional learning branch ψ_c and the re-balancing branch ψ_r. Two



(a) CIFAR-10-LT (b) CIFAR-100-LT

Fig. 3. Class sampling distribution P_k obtained using different values of hyperparameter γ for the re-balanced sampler. We present the class sampling distribution on (a) CIFAR-10-LT with an imbalance ratio of 100 and (b) CIFAR-100-LT with an imbalance ratio of 100.

separate mini-batches are constructed using a uniform sampler and a re-balanced sampler to train each branch network. The uniform sampler draws a sample with an equal probability $P = 1/N$, where N is the cardinality of the training set. The re-balanced sampler draws a sample from the class k with a probability

$$P_k = \frac{w_k}{\sum_{k=1}^{K} w_k} \tag{1}$$

$$w_k = \left(\frac{N_{max}}{N_k}\right)^{\frac{1}{\gamma}}, \tag{2}$$

where γ is the hyperparameter, N_k is the sample size of class k, and N_{max} is the maximum sample size for all the classes. The hyperparameter γ determines the class distribution $P_k(k = 1, \dots, K)$ used for batch-sampling in the re-balanced sampler. As γ increases, the class distribution approaches a uniform distribution, which means that the classes of the samples in the batch are evenly distributed. As γ decreases, the re-balanced sampler samples the minority class data more frequently. See Fig. 3 for the class sampling distribution for different values of γ on CIFAR-10-LT and CIFAR-100-LT datasets. Two samples x_c and x_r are generated by the uniform and re-balanced samplers, respectively, and then are transformed to $\hat{x}_c$ and $\hat{x}_r$ by the proposed bilateral mixup operation. The transformed samples are then fed to the shared backbone network $\phi(\cdot)$ followed by two subsequent branch networks $\psi_c(\cdot)$ and $\psi_r(\cdot)$, i.e.,

$$z_c = \psi_c(\phi(\hat{x}_c)) \tag{3}$$

$$z_r = \psi_r(\phi(\hat{x}_r)), \tag{4}$$

where $z_c = [z_{c,1}, \dots, z_{c,K}]^T$ and $z_r = [z_{r,1}, \dots, z_{r,K}]^T$ are the K-dimensional logits.

3.2. Bilateral mixup augmentation

Recall that the original mixup augmentation [22] generates the sample $(\tilde{x}, \tilde{y})$ by taking a convex combination of two samples (x_i, y_i) and (x_j, y_j), i.e.,

$$\lambda \sim Beta(\alpha, \alpha) \tag{5}$$

$$\begin{pmatrix}\tilde{x}\\ \tilde{y}\end{pmatrix} = \lambda \begin{pmatrix}x_i\\ y_i\end{pmatrix} + (1-\lambda)\begin{pmatrix}x_j\\ y_j\end{pmatrix}, \tag{6}$$

where $Beta(\cdot,\cdot)$ denotes a beta distribution and α denotes the hyperparameter for the beta distribution. The bilateral mixup augmentation simply takes two samples (x_c, y_c) and (x_r, y_r) from the uniform sampler and the re-balanced sampler, respectively, and combines them with different ratios, i.e.,

$$\lambda \sim Beta(\alpha, \alpha) \tag{7}$$

$$\lambda_c = \max(\lambda, 1-\lambda) \tag{8}$$

$$\lambda_r = \min(\lambda, 1-\lambda) \tag{9}$$

$$\begin{pmatrix}\hat{x}_c\\ \hat{y}_c\end{pmatrix} = \lambda_c \begin{pmatrix}x_c\\ y_c\end{pmatrix} + (1-\lambda_c)\begin{pmatrix}x_r\\ y_r\end{pmatrix} \tag{10}$$

$$\begin{pmatrix}\hat{x}_r\\ \hat{y}_r\end{pmatrix} = \lambda_r \begin{pmatrix}x_c\\ y_c\end{pmatrix} + (1-\lambda_r)\begin{pmatrix}x_r\\ y_r\end{pmatrix}. \tag{11}$$

Two bilateral mixup samples $(\hat{x}_c, \hat{y}_c)$ and $(\hat{x}_r, \hat{y}_r)$ are fed to the conventional learning branch and the re-balancing branch, respectively. As seen in (7)–(11), our novel *class distribution-aware mixup strategy* combines samples from two different samplers to improve the representation of long-tailed class data. Note that the conventional mixup [22] cannot improve the representation of minority class data since it does not take into account the distribution of majority and minority classes, and thus the mixup operation would have affected the majority class samples only.

3.3. Class-wise temperature scaling

During the training period, class-wise temperature scaling is applied to the logits z_c and z_r

$$\hat{p}_{c,k} = \exp\left(\frac{z_{c,k}}{\mathcal{T}_k}\right) \Big/ \sum_{k=1}^{K} \exp\left(\frac{z_{c,k}}{\mathcal{T}_k}\right) \tag{12}$$

$$\hat{p}_{r,k} = \exp\left(\frac{z_{r,k}}{\mathcal{T}_k}\right) \Big/ \sum_{k=1}^{K} \exp\left(\frac{z_{r,k}}{\mathcal{T}_k}\right), \tag{13}$$

where $\mathcal{T}_k$ denotes the temperature parameter. The parameter $\mathcal{T}_k$ is set differently per class to reduce the classifier bias caused by the long-tailed class distribution. The parameter $\mathcal{T}_k$ is given by

$$\mathcal{T}_k = \left(\frac{\max(\mathcal{B}_{1:K})}{\mathcal{B}_k}\right)^{\frac{1}{\eta}} \tag{14}$$

$$\mathcal{B}_k = \epsilon \frac{N_k}{N_{max}} + (1-\epsilon), \tag{15}$$

where η and ϵ are hyperparameters, $\mathcal{B}_{1:K} = \{\mathcal{B}_1, \dots, \mathcal{B}_K\}$, N_k is the number of samples in the kth class, and N_{max} is the maximum sample size for all classes. Our class-wise temperature scaling compensates for a classifier bias by encouraging the model to favor minority classes over majority classes during the training phase. In (12) and (13), we scale the logit values for the kth class by the temperature parameter $\mathcal{T}_k$. As $\mathcal{T}_k$ increases, the preference for the kth class increases. Therefore, in (14) and (15), we set the temperature parameter $\mathcal{T}_k$ inversely proportional to the number of samples in the kth class. This leads to a lower temperature value $\mathcal{T}_k$ for the majority classes and a higher temperature value $\mathcal{T}_k$ for the minority classes. Note that prior studies proposed a bias-based logit adjustment method to compensate for the classifier bias [3,4,46]. These methods are not suitable for use with the proposed

bilateral mixup because the predictions $\hat{p}_c$ and $\hat{p}_r$ are obtained from mixed input images with soft label values.

The loss function used to train the DBN-Mix is composed of the following terms

$$\mathcal{L}_{total} = \frac{1}{2}\mathcal{L}(\hat{p}_c, \hat{y}_c) + \frac{1}{2}\mathcal{L}(\hat{p}_r, \hat{y}_r), \tag{16}$$

where $\hat{p}_c = [\hat{p}_{c,1}, \dots, \hat{p}_{c,K}]^T$, $\hat{p}_r = [\hat{p}_{r,1}, \dots, \hat{p}_{r,K}]^T$, and $\mathcal{L}$ denotes the cross-entropy loss, $\mathcal{L}(p, y) = -\sum_{k=1}^{K} y_k \log p_k$. Note that the entire network is trained in an end-to-end manner.

3.4. Model inference

During the inference phase, a single test image x is fed into the two branch networks. The outputs from the dual-branch networks are then combined using equal weights

$$z = \frac{1}{2}(\psi_c(\phi(x)) + \psi_r(\phi(x))). \tag{17}$$

Finally, the softmax function is applied to the combined logit z without class-wise temperature scaling.

3.5. Application to single-branch network

While bilateral mixup is primarily intended for DBN architecture, it can also be applied to SBN with a common single-branch structure. Suppose that the SBN generates classification output $\hat{p}$ for a given input $\hat{x}$. The input to the SBN is obtained by applying bilateral mixup augmentation to two samples (x_c, y_c) and (x_r, y_r) generated by the uniform sampler and the re-balanced sampler, respectively

$$\lambda \sim Beta(\alpha, \alpha) \tag{18}$$

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \lambda \begin{pmatrix} x_c \\ y_c \end{pmatrix} + (1-\lambda) \begin{pmatrix} x_r \\ y_r \end{pmatrix}. \tag{19}$$

The resulting samples $(\hat{x}, \hat{y})$ are used to train the SBN through the loss function $\mathcal{L}_{total} = \mathcal{L}(\hat{p}, \hat{y})$. This method is referred to as *SBN-Mix*.

4. Experiments

In this section, we evaluate the proposed method using four datasets for long-tailed visual recognition task: CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018. We also present an ablation study to evaluate the contribution of the components in the DBN-Mix.

4.1. Long-tailed recognition datasets

4.1.1. Long-tailed CIFAR

The long-tailed versions of the CIFAR datasets have been artificially generated based on the original CIFAR-10 and CIFAR-100 datasets [4]. The degree of class imbalance in these datasets was specified by the imbalance ratio $\mu = \frac{\max_i N_i}{\min_j N_j}$, where N_k be the number of training samples in the kth class. We tried several imbalance ratios from $\mu \in \{10, 20, 50, 100, 200\}$ in the experiments. These imbalanced datasets are denoted as CIFAR-10-LT (μ) and CIFAR-100-LT (μ), where μ represents the imbalance ratio.

4.1.2. Long-tailed ImageNet

The original ImageNet [1] is one of the largest image recognition datasets, which contains 1280K training images and 50K test images with 1000 categories. Following [54], we built a long-tailed version of the ImageNet dataset, which contains 115.8K training images. With this modification, the largest class size becomes 1280 and the smallest one becomes 5.

4.1.3. iNaturalist 2018

The iNaturalist 2018 dataset [55] is a large real-world dataset that exhibits a naturally imbalanced distribution of classes. This dataset contains 437.5K training images with 8142 categories. Any modification was not applied to adjust the imbalance ratio.

4.2. Evaluation metrics

The performance of the training methods was measured using the top-1 accuracy metric. The test and validation sets have balanced sample sizes for all the classes. Following [54], we group the data samples into *Many*, *Medium* and *Few* classes according to their size, where *Many* denotes classes with more than 100 samples, *Medium* denotes classes with 20 to 100 samples, and *Few* denotes classes with less than 20 samples.

4.3. Candidate methods

The following baseline algorithms were compared with our method; standard cross-entropy training, Focal loss [40], mixup [22], LDAM-DRW [3], M2 m [17], Remix [19], cRT [23], LWS [23], τ-normalized [23], BBN [8], Meta-weight net [41], MCW + Focal [45] (MCW combined with Focal loss), MetaSAug + LDAM [15] (MetaSAug combined with LDAM loss), Balanced Softmax [56], PaCo [57], MiSLAS [6], Breadcrumb [48], LDAM-DRW + SAFA [50] (LDAM-DRW combined with SAFA), LDAM-DRS + SAFA [50] (LDAM-DRS combined with SAFA), RIDE (4 experts) [5], and RIDE (3 experts) + CMO [20] (RIDE combined with CMO), RISDA [51]. Unless otherwise noted, we used four expert configuration for RIDE.

4.4. Implementation details

In this section, we provide the detailed experimental setups. Table 1 presents the experimental setup used in our experiments.

4.4.1. Long-tailed CIFAR

For both CIFAR-10-LT and CIFAR-100-LT datasets, ResNet-32 [58] was trained with a batch size of 128. We used the same backbone network for all long-tailed recognition methods considered. We used a stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and weight decay of 2×10^{-4}. Weight updates were performed over 200 epochs. We considered two learning rate scheduling strategies for long-tailed CIFAR. Following the learning rate scheduling in [3], we set the initial learning rate to 0.1 and decayed the learning rate by a factor of 0.1 at 120 and 160 epochs, respectively. In addition, following the longer learning rate scheduling in [20,57], we trained the model over 400 epochs and used a learning rate decay of 0.1 at 240 and 320 epochs. The results of this longer schedule are presented in Section 4.7.3. We set $\gamma = \infty$ for the re-balanced sampler.[1] We chose the optimal parameter value of α from the set {0.2, 0.5, 0.7, 1.0, 1.2, 1.5} and that of ϵ from the set $\{0.25, 0.4, 0.6, 0.8\}$ for both CIFAR-10-LT and CIFAR-100-LT. The best temperature parameter η was also chosen from the set $\{2, 3, 5, 7\}$ for CIFAR-10-LT and from the set $\{3, 5, 7, 10\}$ for CIFAR-100-LT. Finally, the parameters used for CIFAR-10-LT were set to $\alpha = 1.0, \eta = 3$, and $\epsilon = 0.6$ and those for CIFAR-100-LT were set to $\alpha = 1.0, \eta = 7$, and $\epsilon = 0.6$. We applied standard data augmentation methods including horizontal flipping and random cropping [58].

[1] By setting γ to infinity, the re-balanced sampler becomes a class-balanced sampler. Therefore, we can simply implement this setup by first selecting a class of a sample based on a uniform distribution and drawing a sample that belongs to that class.

Table 1
Detailed experimental configuration for our experiments.

Dataset	Common setting				Hyperparameters for DBN-Mix			
	Batch size	Initial learning rate	Weight decay	Momentum	α	γ	η	ϵ
CIFAR-10-LT	128	0.1	2×10^{-4}	0.9	1.0	∞	3	0.6
CIFAR-100-LT	128	0.1	2×10^{-4}	0.9	1.0	∞	7	0.6
ImageNet-LT (Standard)	256	0.2	2×10^{-4}	0.9	0.2	∞	9	0.4
ImageNet-LT (Extended)	256	0.2	2×10^{-4}	0.9	0.2	∞	9	0.2
iNaturalist 2018	256	0.1	1×10^{-4}	0.9	0.2	∞	10	2×10^{-2}

Table 2
Top-1 test accuracy (%) of ResNet-32 evaluated on CIFAR-10-LT and CIFAR-100-LT with various imbalance ratios.

Dataset	CIFAR-10-LT					CIFAR-100-LT				
Imbalance Ratio	200	100	50	20	10	200	100	50	20	10
Cross-Entropy	65.87	70.14	74.94	82.44	86.18	34.70	38.46	44.02	51.06	55.73
Focal loss [40]	65.29	70.38	76.71	82.76	86.66	35.62	38.41	44.32	51.95	55.78
Mixup[a] [22]	–	73.06	77.82	–	87.10	–	39.54	44.99	–	58.02
LDAM-DRW [3]	–	77.03	–	–	88.16	38.45	42.89	47.97	52.99	58.78
LDAM + M2m [17]	–	79.10	–	–	87.50	–	43.50	–	–	57.60
LDAM-DRW + SAFA [50]	77.53	80.48	83.57	86.38	88.94	42.47	46.04	50.02	55.88	59.11
Remix [19]	–	79.76	–	–	89.02	–	46.77	–	–	61.23
BBN[a] [8]	–	79.82	82.18	–	88.32	–	42.56	47.02	–	59.12
Meta-weight net[b] [41]	67.20	73.57	79.10	84.45	87.55	36.62	41.61	45.66	53.04	58.91
MCW + Focal[b] [45]	74.43	78.90	82.88	86.10	88.37	39.34	44.70	50.08	55.73	59.59
MetaSAug + LDAM [15]	77.35	80.66	84.34	88.10	89.68	43.09	48.01	52.27	57.53	61.28
MiSLAS [6]	–	82.10	85.70	–	90.00	–	47.00	52.30	–	63.20
GCL [26]	79.03	82.68	85.48	–	–	44.88	48.71	53.55	–	–
RIDE [5]	–	–	–	–	–	–	49.10	–	–	–
RIDE + CMO [20]	–	–	–	–	–	–	50.00	53.00	–	60.20
RISDA [51]	–	–	–	–	–	44.76	50.16	53.84	58.67	62.38
SBN-Mix	69.87	76.33	81.04	86.91	89.84	40.30	45.07	50.39	57.28	62.37
DBN-Mix	**79.58**	**83.47**	**86.82**	**89.11**	**90.87**	**46.21**	**51.04**	**54.93**	**61.07**	**64.98**

[a] The entries are taken from the results reported in [8].
[b] The entries are taken from the results reported in [45].

4.4.2. Long-tailed ImageNet and iNaturalist 2018

In ImageNet-LT, ResNet-50 [58] was used with a batch size of 256. Regarding SGD, the momentum, initial learning rate, and weight decay were set to 0.9, 0.2, and 2×10^{-4}, respectively. In the previous works, two different learning rate schedules were used to train the existing methods. For the *standard schedule*, the initial learning rate decayed by the factor of 0.1 at 60 and 80 epochs, and for the *extended schedule*, the learning rate decayed at 120 and 160 epochs. In our experiments, we presented the results obtained using both schedules. Following the previous study [58], we applied horizontal flipping, resizing to 256×256, and random cropping to 224×224 for data augmentation. In the inference step, the 224×224 patch was cropped from the center of the image and used as an input sample.

In iNaturalist 2018, ResNet-50 [58] was also used. The momentum, initial learning rate, and weight decay were set to 0.9, 0.1, and 1×10^{-4}, respectively. The learning rate decayed by a factor of 10 at 120 and 160 epochs. We applied the same data augmentation used for ImageNet-LT.

Because the hyperparameter ϵ affects a margin for the classifier, it should be determined by considering the class distribution of the dataset. The hyperparameter ϵ should be determined separately for each dataset to ensure sufficient margin for each class. Since the number of classes in iNaturalist 2018 is larger than the number of classes in ImageNet-LT, smaller values of ϵ are preferred in iNaturalist 2018. Taking this into account, we searched for the best value of ϵ from the set $\{0.05, 0.1, 0.2, 0.4\}$ for ImageNet-LT and from the set $\{0.005, 0.01, 0.02, 0.05\}$ for iNaturalist 2018. We also chose the optimal parameter value of η from the set $\{7, 8, 9, 10, 12\}$ and the optimal parameter value of α from the set $\{0.1, 0.2, 0.5, 0.7, 1.0\}$ for both ImageNet-LT and iNaturalist 2018 datasets. Finally, we set $\alpha = 0.2, \eta = 9$, and $\epsilon = 0.2$ for ImageNet-LT and $\alpha = 0.2, \eta = 10$, and $\epsilon = 2\times10^{-2}$ for iNaturalist 2018. For both cases, we set $\gamma = \infty$ for the re-balanced sampler.

4.5. Experimental results

4.5.1. CIFAR-LT

Table 2 presents the classification accuracy of the proposed DBN-Mix method, compared with the existing methods on the CIFAR-10-LT and CIFAR-100-LT datasets. The proposed method outperforms existing methods by significant margins for all imbalance ratio configurations. Among these methods, Remix, MiSLAS, RISDA, and SAFA employed data augmentation strategies for long-tailed visual recognition tasks, which can be compared with that of DBN-Mix. Remix used a mixup strategy that assigned a higher mixing factor to labels associated with minority class samples. MiSLAS employed a combination of mixup augmentation and label-aware smoothing to improve confidence calibration. While these methods did not account for class distribution for their mixup operations, the proposed mixup strategy considered a class distribution to enhance the representation of long-tailed data. RISDA transferred the variance information from other classes to minority classes. SAFA employed a sample-adaptive augmentation to transfer useful class-independent information from the majority classes to the minority classes. While RISDA and SAFA required a complex process to transfer useful information between classes, the proposed bilateral mixup provided a simple yet effective way to generate augmented samples for DBN-Mix. In particular, for CIFAR-10-LT (100) (i.e., an imbalance ratio of 100), DBN-Mix achieves a 1.37% better performance than MiSLAS. Despite these performance improvements, DBN-Mix does not require two-stage training like MiSLAS. The performance gain of DBN-Mix is even higher on CIFAR-100-LT (100). DBN-Mix outperforms MiSLAS by 4.04%. The proposed method outperforms RIDE (four experts) and RIDE (three experts) + CMO by 1.94% and 1.04%, respectively. Though the proposed method is based on two branch networks, it outperforms multiple expert networks with more expert branches.

Table 3
Top-1 test accuracy (%) of ResNet-50 evaluated on ImageNet-LT.

Method	All	Many	Medium	Few
Cross-Entropy	41.6	64.0	33.9	5.8
Cross-Entropy[a]	44.6	–	–	–
LDAM-DRW[a]	48.8	–	–	–
RISDA	50.7	–	–	–
LWS	47.7	57.1	45.2	29.3
LWS[a]	52.0	62.9	49.8	31.6
MiSLAS[a]	52.7	61.7	51.3	35.8
RIDE	55.4	66.2	52.3	36.5
RIDE + CMO	56.2	66.4	**53.9**	35.6
DBN-Mix	55.9	67.2	53.4	35.9
DBN-Mix[a]	**56.6**	**67.9**	53.2	**38.0**

[a] The results are obtained using a longer training schedule (2×) following the settings in [6].

Table 4
Top-1 test accuracy (%) on iNaturalist 2018. All methods were trained for 200 epochs. ResNet-50 was used as a backbone network. A longer training schedule (2×) was not used on iNaturalist 2018 dataset. The performance of other methods is taken from the results reported in [6,48,50].

Method	All	Many	Medium	Few
Cross-Entropy	61.7	72.2	63.0	57.2
LDAM-DRW	68.0	–	–	–
BBN	69.3	49.4	70.8	65.3
LDAM-DRS + SAFA	69.8	–	–	–
Remix	70.5	–	–	–
Breadcrumb	70.3	–	–	–
cRT	70.2	**74.2**	71.1	68.2
LWS	70.9	72.8	71.6	69.8
MiSLAS	71.6	73.2	72.4	70.4
RIDE	72.6	70.9	72.4	73.1
RIDE + CMO	72.8	68.7	72.6	73.1
DBN-Mix	**74.7**	73.0	**75.6**	**74.7**

4.5.2. ImageNet-LT

Table 3 presents the top-1 accuracies of several methods evaluated on the ImageNet-LT dataset. We observe that the proposed DBN-Mix also outperforms the existing methods. In the standard training schedule, DBN-Mix performs 5.2% better than RISDA, a state-of-the-art augmentation method that requires additional inter-class information. DBN-Mix achieves 8.2% and 0.8% better performance than LWS and RIDE, respectively. While both LWS and RIDE require a two-stage training process, DBN-Mix trains the entire dual-branch network end-to-end. For an extended training schedule, DBN-Mix achieves a performance gain of 3.9% over MiSLAS and 4.6% over LWS. DBN-Mix improves the performance for both learning schedules, which shows that our method achieves a consistent performance improvement on long-tailed recognition. Table 3 shows that after a longer training schedule (2×), the performance improves for *Many* and *Few* classes but slightly degrades for *Medium* classes. We used the different configurations $\epsilon = 0.4$ and $\epsilon = 0.2$ for the 1× and 2× training schedules, respectively, which resulted in different behavior in the bias correction, as shown in Table 3.

4.5.3. iNaturalist 2018

In Table 4, the performance of DBN-Mix is evaluated on iNaturalist 2018 dataset. DBN-Mix achieves the best classification accuracy among competitors. DBN-Mix achieves a 4.9% performance gain over LDAM-DRS + SAFA, a knowledge transfer-based method. Our method achieves a 2.1% performance gain over RIDE and 1.9% performance gain over RIDE (3 experts) + CMO [20], the CutMix-based state-of-the-art method. In particular, the performance gain of DBN-Mix over other methods is significant for the 'few' category, which shows that the proposed ideas are effective in recognizing the minority class samples. Following the convention of previous studies [3,56], we configured the parameters of DBN-Mix to achieve the best performance in the *All Class* category on the class-balanced test set. So the performance of the *Many* class category would have been traded for the *Few* class category to achieve the best performance in the *All Class* category. Please note that we can improve the performance of the *Many* category at the expense of the *Few* category through hyperparameter tuning. In Section 4.6.4, we presented the experimental results on CIFAR-100-LT (100) in this regard.

4.6. Ablation study

4.6.1. Contributions of key ideas

In Table 5, we analyze the impact of the two main ideas on overall performance: (1) bilateral mixup augmentation and (2) class-wise temperature scaling. The DBN structure used in BBN [8] is selected as the baseline. The vanilla SBN trained with cross-entropy loss is also selected as the baseline.

Bilateral mixup offers a performance gain of 5.56% over the DBN baseline on CIFAR-100-LT (100). This gain is larger than the performance gain of 2.12% achieved by the conventional mixup [22]. Temperature scaling offers a performance gain of 3.03%. A considerable performance improvement of 9.99% is achieved when bilateral mixup and temperature scaling are applied together. The performance gain of the proposed method increases as the imbalance ratio increases, which implies that the proposed DBN-Mix can handle severely imbalanced class distributions better than the other methods.

When we apply the bilateral mixup augmentation to the SBN baseline, classification accuracy improves by 5.64%. We compare this result to the case where conventional mixup augmentation [22] is applied to SBN. The performance gain achieved by the conventional mixup is only 1.08%, demonstrating the superiority of the proposed bilateral mixup augmentation on the long-tailed recognition problem. Temperature scaling does not offer a large gain (i.e., 2.79%) for SBN without bilateral mixup. However, when bilateral mixup augmentation and temperature scaling are used together, the performance gain increases dramatically to 6.61%, which shows that temperature scaling successfully complements the bilateral mixup.

4.6.2. Performance versus hyperparameters

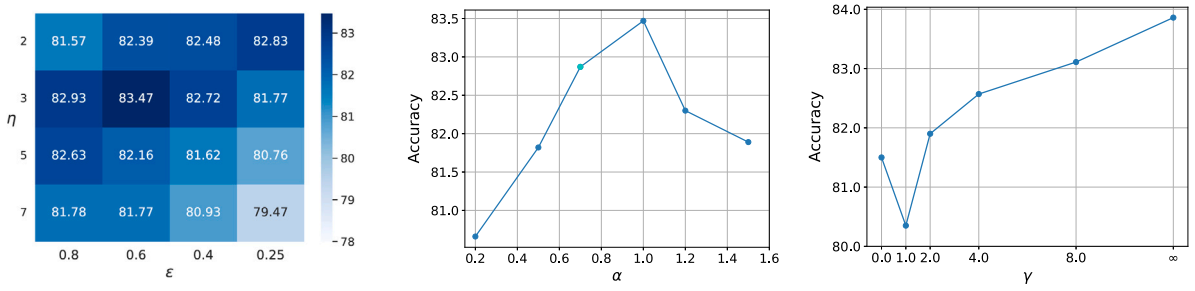
Fig. 4(a) and Fig. 5(a) present the performance of DBN-Mix as a function of the parameters η and ϵ evaluated on CIFAR-10-LT (100) and CIFAR-100-LT (100), respectively. A larger ϵ or smaller η enhances the effect of temperature scaling. We empirically searched for the best value of ϵ from the set $\{0.25, 0.4, 0.6, 0.8\}$ for both CIFAR-LT datasets. For the best value of η, we searched from the set $\{2, 3, 5, 7\}$ for CIFAR-10-LT and from the set $\{3, 5, 7, 10\}$ for CIFAR-100-LT. Through extensive experiments, we find that the setups $(\eta, \epsilon) = (3, 0.6)$ and $(\eta, \epsilon) = (7, 0.6)$ achieve the best performances for CIFAR-10-LT and CIFAR-100-LT, respectively.

Fig. 4(b) and Fig. 5(b) present the performance with respect to α, the parameter of the beta distribution used in the bilateral mixup. We tried different values of α from the set $\{0.2, 0.5, 0.7, 1.0, 1.2, 1.5\}$. The best performance was achieved at $\alpha = 1.0$ for both the CIFAR-10-LT and CIFAR-100-LT datasets and at $\alpha = 0.2$ for the ImageNet-LT and iNaturalist 2018 datasets.

Finally, Fig. 4(c) and Fig. 5(c) provide the performance of DBN-Mix as a function of γ used in the re-balanced sampler. As γ increases above 1.0, the distribution of samples generated by the re-balanced sampler approaches a class-balanced distribution. This change in sampling distribution serves to mitigate the overfitting problem caused by the lack of minority class samples and thereby improve the accuracy as shown in Fig. 4(c) and Fig. 5(c). We used the class-balanced sampling $\gamma = \infty$, which offers the best performance.

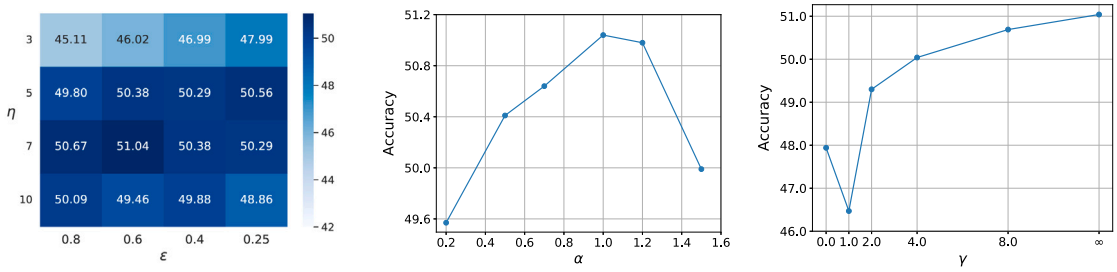
Table 5
Ablation study of each main component in DBN-Mix. All methods were trained for 200 epochs on CIFAR-100-LT with various imbalance ratios. ResNet-32 was used as a backbone network. A longer training schedule (2×) was not used on CIFAR-100-LT dataset.

Method	Main components		Imbalance ratio		
	Bilateral mixup	Temperature scaling	100	50	10
Vanilla single branch net.			38.46	44.02	55.73
Single branch net. + mixup			39.54	44.99	58.02
Dual branch net. + mixup			43.17	48.18	61.66
Single branch net.		✓	41.25	45.64	58.36
	✓		44.10	49.73	61.98
	✓	✓	45.07	50.39	62.37
Dual branch net.			41.05	46.93	60.82
		✓	44.08	50.64	62.98
	✓		46.61	51.42	63.59
	✓	✓	**51.04**	**54.93**	**64.98**



(a) (b) (c)

Fig. 4. Performance versus hyperparameters: (a) η and ϵ for temperature scaling, (b) α for bilateral mixup augmentation, and (c) γ for re-balanced sampler. The performance was evaluated on CIFAR-10-LT (100) dataset. ResNet-32 was used as a backbone network.



(a) (b) (c)

Fig. 5. Performance versus hyperparameters: (a) η and ϵ for temperature scaling, (b) α for bilateral mixup augmentation, and (c) γ for re-balanced sampler. The performance was evaluated on CIFAR-100-LT (100) dataset. ResNet-32 was used as a backbone network.

Table 6
Performance of temperature scaling methods compared with other re-weighting methods [3,56]. ResNet-32 was used as a backbone network. We conducted the experiments using a dual-branch network with bilateral mixup on CIFAR-100-LT (100) for all methods.

Method	All	Many	Medium	Few
Cross-Entropy	46.61	72.20	46.00	13.40
LDAM-DRW [3]	46.80	**72.94**	46.31	15.00
Balanced Softmax [56]	47.46	51.21	50.22	**38.29**
Learnable Temperature Scaling	50.91	66.89	54.34	28.28
Temperature Scaling	**51.04**	66.03	**55.49**	28.36

Table 7
Performance of DBN-Mix obtained using different sets of hyperparameters. ResNet-32 was used as a backbone network.
We conducted the experiments using a dual-branch network on CIFAR-100-LT (100).

Method	Hyperparameters		All	Many	Medium	Few
	ϵ	η				
Cross-Entropy	–	–	46.61	**72.20**	46.00	13.40
DBN-Mix	0.1	12	48.05	71.42	50.67	17.71
	0.2	11	48.95	70.50	51.39	20.28
	0.3	10	49.52	69.32	52.25	23.23
	0.4	9	49.89	68.69	52.84	24.54
	0.5	8	50.42	67.36	53.35	26.34
	0.6	7	**51.04**	66.03	**55.49**	**28.36**

4.6.3. *Temperature scaling methods compared with re-weighting methods*

Table 6 compares the performance of the proposed temperature scaling with other re-weighting methods and the learnable temperature scaling. The initial values of η and ϵ were obtained from $10 \cdot \text{sigmoid}(t)$ and $\text{sigmoid}(t')$, respectively, where t and t' were sampled from Gaussian distribution $N(1, 1)$. First, we compare our temperature scaling methods with other popular re-weighting methods: vanilla cross-entropy training, LDAM-DRW [3], and Balanced Softmax [56].

Table 8
Performance of DBN-Mix evaluated at different branches and for different class groups. CIFAR-100-LT (100) dataset was used for evaluation. ResNet-32 was used as a backbone network.

Method	Branch	All	Many	Medium	Few
BBN	Re-balancing branch	35.30	40.41	46.61	16.13
	Conventional branch	38.52	65.78	37.10	8.37
	Final	42.95	65.86	45.38	13.40
DBN-Mix	Re-balancing branch	47.16	56.37	54.18	28.23
	Conventional branch	48.03	**66.55**	50.08	24.06
	Final	**51.04**	66.03	**55.49**	**28.36**

We apply them to the same dual-branch network setup we use for our method. Although LDAM-DRW [3] and Balanced Softmax [56] achieve better performance in some categories, the proposed class-wise temperature scaling outperforms both by 4.24% and 3.58% in overall accuracy. This indicates that the proposed temperature scaling method offers more effective control of classifier bias than other re-weighting methods. Learnable temperature scaling is a class-wise temperature scaling that uses η and ϵ as learnable parameters. The proposed temperature scaling yields slightly better performance than the learnable temperature scaling.

4.6.4. Performance trade-offs between many and few categories for different hyperparameters

Following the strategy of the previous studies [3,56], we configured the parameters of DBN-Mix to achieve the best performance in the *All Class* category on the class-balanced test set. So the performance of the *Many* class category would have been traded for the *Few* class category to achieve the best performance in the *All Class* category. Please note that we can improve the performance of the *Many* category at the expense of the *Few* category through hyperparameter tuning. Table 7 shows the performance results obtained using different sets of hyperparameters on CIFAR-100-LT (100).

4.7. Analysis and discussion

4.7.1. Performance evaluated at different branches and for different class groups

Table 8 reports a thorough analysis of the top-1 accuracy evaluated at different branches (*Re-balancing branch, Conventional branch, and Final output*) and for three class groups (*Many*, *Medium*, and *Few*). We measured the output accuracy for each branch. *Many* denotes the set of majority class samples and *Few* denotes the set of minority class samples.

DBN-Mix offers significant performance improvements over the BBN baseline at both points. A performance gain of 11.86% was achieved for the re-balancing branch and a performance gain of 9.51% was achieved for the conventional learning branch. DBN-Mix maintains strong performance at all points for the *Many* group. DBN-Mix also achieves significant performance gains over BBN for both the *Medium* and *Few* groups. Even for the *Few* group in the conventional branch, BBN achieves a performance of only 8.37%, whereas our DBN-Mix achieves a much higher performance of 24.06%. Compared to the cumulative learning used in BBN, the proposed class distribution-aware augmentation successfully mitigates the overfitting problem for the minority classes within each branch.

4.7.2. Representation learning performance versus classifier learning performance

A study published in [23] showed that evaluating the performance of a model separately in terms of representation learning (RL) performance and classifier learning (CL) performance provides useful insights into understanding the behavior of its key components. Tables 9 and 10 show the RL and CL performances of DBN-Mix achieved by adding each idea individually. For RL performance, a classifier trained with cRT [23] at the second stage was used. To evaluate the CL performance, a backbone network trained with conventional mixup augmentation [22] for 200 epochs was used.

Table 9
Analysis of representation learning performance. The performance was evaluated on CIFAR-100-LT (100) dataset. ResNet-32 was used as a backbone network.

Method	Representation learning		Accuracy
	Bilateral mixup	Temperature scaling	
Dual-branch net.			38.78
		✓	40.42
	✓		45.57
	✓	✓	**46.25**

Table 10
Analysis of classifier learning performance of DBN-Mix. The performance was evaluated on CIFAR-100-LT (100) dataset. ResNet-32 was used as a backbone network.

Method	Classifier learning		Accuracy
	Bilateral mixup	Temperature scaling	
Dual-branch net.			43.23
		✓	45.86
	✓		42.67
	✓	✓	**46.52**

Table 11
Top-1 test accuracy (%) of ResNet-32 on CIFAR-100-LT. Performance of DBN-Mix evaluated for the imbalance ratio from $\mu \in \{10, 50, 100\}$. Following the experimental setup used in [20], each method trains the model for 400 epochs with AutoAugment [59].

Method	Imbalance ratio		
	100	50	10
Balanced Softmax [56]	50.8	54.2	63.0
PaCo [57]	52.0	56.0	64.2
CMO + Balanced Softmax [20]	51.7	56.7	65.3
DBN-Mix	**54.3**	**57.7**	**66.4**

The bilateral mixup gives a significant performance improvement of 6.79% in terms of the RL performance. In contrast, the bilateral mixup only marginally improves CL performance. It can be seen that combining samples from two different samplers in the bilateral mixup serves to alleviate overfitting for the minority classes and thus improve the RL. We also see that the temperature scaling improves the RL performance by 1.64% and the CL performance by 2.63%. This shows that compensating for bias caused by the class imbalance improves both RL and CL performance.

4.7.3. Training on longer schedule with strong augmentation method

Following the experimental setup used in [20,57], we trained the model over 400 epochs with AutoAugment [59] on CIFAR-100-LT. Table 11 shows that the proposed method outperforms the existing methods with strong augmentation by significant margins for all the imbalance ratios considered. In particular, for the imbalance ratio of 100, DBN-Mix outperforms PaCo by 2.3% and Balanced Softmax with CMO by 2.6%. A notable point is that an additional strong augmentation method [59] can significantly improve our bilateral mixup even if our method is much simpler in terms of structure.

4.7.4. T-SNE visualization evaluated at different branches

We visualize the feature vector of the penultimate layer of conventional learning branch and re-balancing branch using T-SNE [60]. The T-SNE visualization method projects a feature vector onto a lower-dimensional embedding space. Following the experimental setup used in Table 2, we train the DBN on CIFAR-10-LT with an imbalance ratio of 100. Fig. 6 shows the feature of the conventional learning branch and the re-balancing branch in DBN trained by BBN and DBN-Mix, respectively. For the features trained by BBN, we can observe that the features

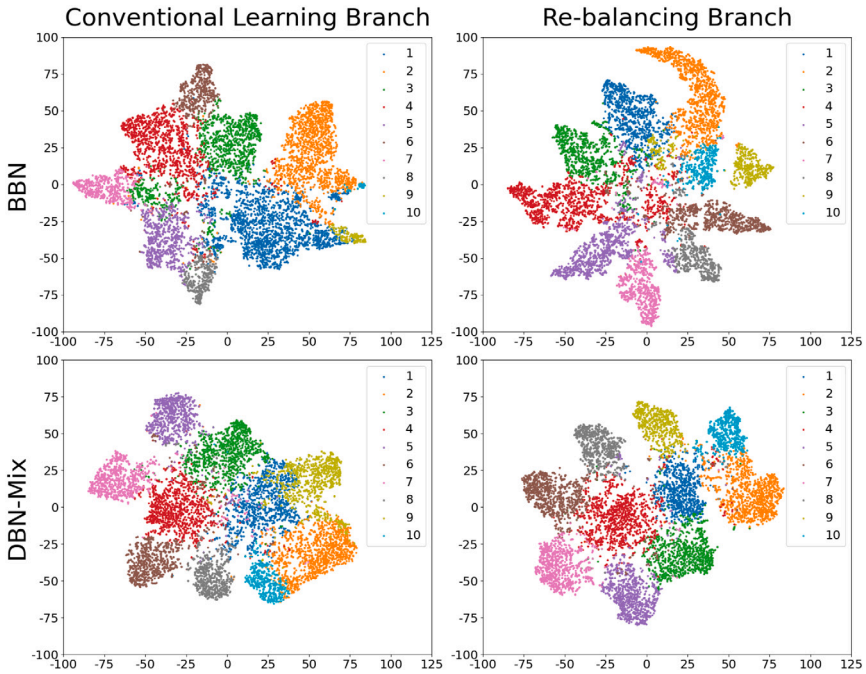


Fig. 6. T-SNE [60] illustrates the feature at the penultimate layer of the conventional learning branch and re-balancing branch. The feature of the conventional learning branch (first column) and the re-balancing branch (second column) in DBN are trained by BBN (first row) and DBN-Mix (second row) [8], respectively.

of each class are entangled in both the conventional learning branch and the re-balancing branch, making it difficult to distinguish them from other classes. DBN-Mix obtains more clear decision boundaries and learns better representations for all classes.

5. Conclusion

In this paper, we studied the problem of training a DNN-based classification model using a dataset with a long-tailed class distribution. We proposed the bilateral mixup augmentation method to prevent the re-balancing branch of the DBN from degrading representation learning. The bilateral mixup achieved this goal simply by training with a convex combination of minority and majority training samples. We also proposed class conditional temperature scaling to compensate for the bias caused by class imbalance. Our experiments on several long-tailed visual recognition datasets confirmed that the proposed DBN-Mix outperformed the DBN baseline and achieved state-of-the-art performance on various benchmarks. In practice, the class bias in the dataset is not simply determined by the number of samples in each class and its degree changes continuously with the model during training. In this sense, the proposed DBN-Mix is limited in its ability to adjust the sampling strategy based on the severity of class bias during the training phase. Therefore, in future work, we would like to explore strategies to adjust the sampling distribution in the re-balanced sampler online accounting for the class bias factors such as data complexity for each class.

Declaration of competing interest

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

Data availability

The authors do not have permission to share data.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[3] K. Cao, C. Wei, A. Gaidon, N. Aréchiga, T. Ma, Learning imbalanced datasets with label-distribution-aware margin loss, in: Advances in Neural Information Processing Systems, 2019, pp. 1565–1576.

[4] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.

[5] X. Wang, L. Lian, Z. Miao, Z. Liu, S. Yu, Long-tailed recognition by routing diverse distribution-aware experts, in: International Conference on Learning Representations, 2021.

[6] Z. Zhong, J. Cui, S. Liu, J. Jia, Improving calibration for long-tailed recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16489–16498.

[7] L. Xiang, G. Ding, J. Han, Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification, in: European Conference on Computer Vision, Springer, 2020, pp. 247–263.

[8] B. Zhou, Q. Cui, X.-S. Wei, Z.-M. Chen, Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9719–9728.

[9] Z. Jan, B. Verma, Multiple strong and balanced cluster-based ensemble of deep learners, Pattern Recognit. 107 (2020) 107420.

[10] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, Y. Zhou, A novel ensemble method for classifying imbalanced data, Pattern Recognit. 48 (5) (2015) 1623–1637.

[11] J.F. Díez-Pastor, J.J. Rodríguez, C.I. García-Osorio, L.I. Kuncheva, Diversity techniques improve the performance of the best imbalance learning ensembles, Inform. Sci. 325 (2015) 98–117.

[12] W. Liu, Z. Wu, Y. Wang, H. Ding, F. Liu, J. Lin, G. Lin, LCReg: Long-tailed image classification with latent categories based recognition, Pattern Recognit. 145 (2024) 109971.

[13] S. Suh, P. Lukowicz, Y.O. Lee, Discriminative feature generation for classification of imbalanced data, Pattern Recognit. 122 (2022) 108302.

[14] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.

[15] S. Li, K. Gong, C.H. Liu, Y. Wang, F. Qiao, X. Cheng, Metasaug: Meta semantic augmentation for long-tailed visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5212–5221.

[16] P. Chu, X. Bian, S. Liu, H. Ling, Feature space augmentation for long-tailed data, in: European Conference on Computer Vision, 2020, pp. 694–710.
[17] J. Kim, J. Jeong, J. Shin, M2m: Imbalanced classification via major-to-minor translation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13896–13905.
[18] Y. Zhang, X.-S. Wei, B. Zhou, J. Wu, Bag of tricks for long-tailed visual recognition with deep convolutional neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 4, 2021, pp. 3447–3455.
[19] H.-P. Chou, S.-C. Chang, J.-Y. Pan, W. Wei, D.-C. Juan, Remix: Rebalanced mixup, in: European Conference on Computer Vision, 2020, pp. 95–110.
[20] S. Park, Y. Hong, B. Heo, S. Yun, J.Y. Choi, 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, 2022, pp. 6887–6896.
[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[22] H. Zhang, M. Cissé, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.
[23] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, in: International Conference on Learning Representations, 2020.
[24] Y.-X. Wang, D. Ramanan, M. Hebert, Learning to model the tail, in: Advances in Neural Information Processing Systems, 2017, pp. 7032–7042.
[25] K. Tang, J. Huang, H. Zhang, Long-tailed classification by keeping the good and removing the bad momentum causal effect, in: Advances in Neural Information Processing Systems, 2020.
[26] M. Li, Y.-m. Cheung, Y. Lu, Long-tailed visual recognition via gaussian clouded logit adjustment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6929–6938.
[27] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: A new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer, 2005, pp. 878–887.
[28] J. Byrd, Z. Lipton, What is the effect of importance weighting in deep learning? in: Proceedings of International Conference on Machine Learning, 2019, pp. 872–881.
[29] L. Shen, Z. Lin, Q. Huang, Relay backpropagation for effective learning of deep convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 467–482.
[30] I. Cordón, S. García, A. Fernández, F. Herrera, Imbalance: Oversampling algorithms for imbalanced classification in R, Knowl.-Based Syst. 161 (2018) 329–341.
[31] T. Zhu, Y. Lin, Y. Liu, Synthetic minority oversampling technique for multiclass imbalance problems, Pattern Recognit. 72 (2017) 327–340.
[32] M. Buda, A. Maki, M.A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Netw. 106 (2018) 249–259.
[33] S. Maldonado, J. López, C. Vairetti, An alternative SMOTE oversampling strategy for high-dimensional datasets, Appl. Soft Comput. 76 (2019) 380–389.
[34] S. Maldonado, C. Vairetti, A. Fernandez, F. Herrera, FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification, Pattern Recognit. 124 (2022) 108511.
[35] J.A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, Inform. Sci. 291 (2015) 184–203.
[36] C. Drummond, R.C. Holte, et al., C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling, in: Workshop on Learning from Imbalanced Datasets II, Vol. 11, 2003, pp. 1–8.
[37] M. Kubat, S. Matwin, et al., Addressing the curse of imbalanced training sets: one-sided selection, in: Proceedings of International Conference on Machine Learning, Vol. 97, No. 1, 1997, p. 179.
[38] P. Soltanzadeh, M.R. Feizi-Derakhshi, M. Hashemzadeh, Addressing the class-imbalance and class-overlap problems by a metaheuristic-based under-sampling approach, Pattern Recognit. (2023) 109721.
[39] M. Koziarski, Radial-based undersampling for imbalanced data classification, Pattern Recognit. 102 (2020) 107262.
[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 2980–2988.
[41] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, D. Meng, Meta-weight-net: Learning an explicit mapping for sample weighting, in: Advances in Neural Information Processing Systems, 2019, pp. 1917–1928.
[42] M. Ren, W. Zeng, B. Yang, R. Urtasun, Learning to reweight examples for robust deep learning, in: Proceedings of International Conference on Machine Learning, 2018, pp. 4334–4343.
[43] K.P. Alexandridis, J. Deng, A. Nguyen, S. Luo, Long-tailed instance segmentation using gumbel optimized loss, in: European Conference on Computer Vision, Springer, 2022, pp. 353–369.
[44] L. Zhao, Z. Shang, J. Tan, M. Zhou, M. Zhang, D. Gu, T. Zhang, Y.Y. Tang, Siamese networks with an online reweighted example for imbalanced data learning, Pattern Recognit. 132 (2022) 108947.
[45] M.A. Jamal, M. Brown, M. Yang, L. Wang, B. Gong, Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7607–7616.
[46] A.K. Menon, S. Jayasumana, A.S. Rawat, H. Jain, A. Veit, S. Kumar, Long-tail learning via logit adjustment, in: International Conference on Learning Representations, 2021.
[47] Y. Xu, Y.-L. Li, J. Li, C. Lu, Constructing balance from imbalance for long-tailed image recognition, in: European Conference on Computer Vision, Springer, 2022, pp. 38–56.
[48] B. Liu, H. Li, H. Kang, G. Hua, N. Vasconcelos, Breadcrumbs: Adversarial class-balanced sampling for long-tailed recognition, in: European Conference on Computer Vision, Springer, 2022, pp. 637–653.
[49] H. Guo, S. Wang, Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15089–15098.
[50] Y. Hong, J. Zhang, Z. Sun, K. Yan, SAFA: Sample-adaptive feature augmentation for long-tailed image classification, in: European Conference on Computer Vision, Springer, 2022, pp. 587–603.
[51] X. Chen, Y. Zhou, D. Wu, W. Zhang, Y. Zhou, B. Li, W. Wang, Imagine by reasoning: A reasoning-based implicit semantic data augmentation for long-tailed classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 1, 2022, pp. 356–364.
[52] Y. Wang, X. Pan, S. Song, H. Zhang, G. Huang, C. Wu, Implicit semantic data augmentation for deep networks, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 12635–12644.
[53] F. Du, P. Yang, Q. Jia, F. Nan, X. Chen, Y. Yang, Global and local mixture consistency cumulative learning for long-tailed visual recognitions, 2023, arXiv preprint arXiv:2305.08661.
[54] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, S.X. Yu, Large-scale long-tailed recognition in an open world, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2537–2546.
[55] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The inaturalist species classification and detection dataset, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8769–8778.
[56] J. Ren, C. Yu, S. Sheng, X. Ma, H. Zhao, S. Yi, H. Li, Balanced meta-softmax for long-tailed visual recognition, in: Advances in Neural Information Processing Systems, 2020.
[57] J. Cui, Z. Zhong, S. Liu, B. Yu, J. Jia, Parametric contrastive learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 715–724.
[58] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[59] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, Autoaugment: Learning augmentation strategies from data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 113–123.
[60] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.

Jae Soon Baik received the B.S. degree from the Department of Mechanical Engineering, Hanyang University, Seoul, Korea, in 2017, and the M.S. degree from the Department of Automotive Engineering, Hanyang University, Seoul, Korea, in 2019. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, Hanyang University. His research interests include machine learning, active learning, long-tailed recognition, and semi-supervised learning.

In Young Yoon received the B.S. degree in electrical engineering from Hanyang University, Seoul, Korea, in 2021. He is currently a pursuing the combined master's and Ph.D. degrees in the Department of Electrical Engineering, Hanyang University. His current research interests include long-tailed recognition, Camera to BEV Transformation, and BEV semantic segmentation.

Jun Won Choi received B.S and M.S. degrees at Seoul National University and Ph. D. degree at University of Illinois at Urbana-Champaign. In 2010, he joined the company Qualcomm (San Diego, USA) where he participated in research on advanced signal processing technology for the next generation wireless communication technology. Since 2013, he has been a faculty member of the Department of Electrical Engineering, Hanyang University. His research area includes signal processing, machine learning, wireless communications, intelligent vehicles, etc. He has served as Associate Editor of IEEE Trans. Vehicular Technology and IEEE Trans. Intelligent Transportation Systems. He is a reviewer of many technical journals including IEEE Trans. Signal Processing, IEEE Trans. Wireless Communications, IEEE Trans. Intelligent Transportation Systems, etc.