Feature Space Augmentation for Long-Tailed Data

Peng Chu[1⋆], Xiao Bian[2⋆], Shaopeng Liu[3], and Haibin Ling[4,1]

[1] Temple University, USA pchu@temple.edu
[2] Google Inc., USA xbian@google.com
[3] GE Research, USA sliu@ge.com
[4] Stony Brook University, USA hling@cs.stonybrook.edu

Abstract. Real-world data often follow a long-tailed distribution as the frequency of each class is typically different. For example, a dataset can have a large number of under-represented classes and a few classes with more than sufficient data. However, a model to represent the dataset is usually expected to have reasonably homogeneous performances across classes. Introducing class-balanced loss and advanced methods on data re-sampling and augmentation are among the best practices to alleviate the data imbalance problem. However, the other part of the problem about the under-represented classes will have to rely on additional knowledge to recover the missing information.
In this work, we present a novel approach to address the long-tailed problem by augmenting the under-represented classes in the feature space with the features learned from the classes with ample samples. In particular, we decompose the features of each class into a class-generic component and a class-specific component using class activation maps. Novel samples of under-represented classes are then generated on the fly during training stages by fusing the class-specific features from the under-represented classes with the class-generic features from confusing classes. Our results on different datasets such as iNaturalist, ImageNet-LT, Places-LT and a long-tailed version of CIFAR have shown the state of the art performances.

1 Introduction

Deep neural networks have shown considerable success in a wide variety of visual recognition tasks. Its effectiveness and generalizability have been well proved by many state-of-the-art work [25, 19, 16, 30] and a wide variety of real-world applications in different industries [6, 49, 3, 12]. However, there is often one underlying condition that each category of interest needs to be well represented.

To quantify the "representativeness" of data can be a challenging problem itself. In practice, it is usually scrutinized using different heuristics, and

⋆ Work was done at GE Research.

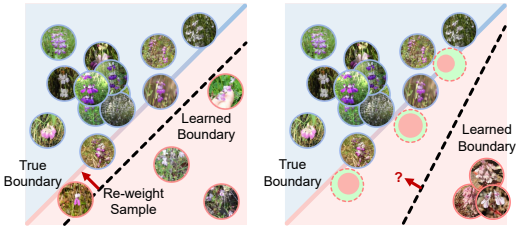

Fig. 1: Left: With limited but well-spread data, the optimal decision boundary search can be recovered by sample re-weighting/loss balancing. Right: Without sufficient sample coverage, the "optimal direction" to move the decision boundary becomes unclear. In this paper, augmented samples are generated to recover the underlying distribution.

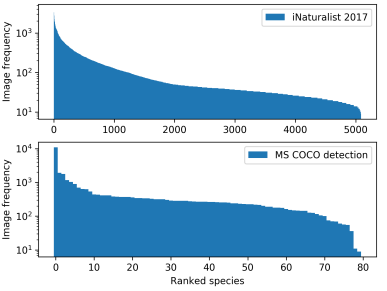

Fig. 2: The sorted sample size of each class from different dataset follows similar long-tailed distribution.

one common criterion could be the balance of a dataset. Indeed, many public datasets are intentionally organized to have the same number of samples from each class [33, 24]. For problems such as segmentation and detection which are hard to ensure exact balanced data, it is always preferable to ensure good data coverage in a way that the rare classes still have sufficient data and are hence well represented [40, 13].

However, real-world visual understanding problems are often fine-grained and long-tailed. To achieve human-level visual understanding, it almost implies the ability to distinguish the subtle differences between fine-grained categories and to robustly handle the presence of rare categories [1]. In fact, these two properties of real-world data usually accompany each other as a large number of fine-grained categories often leads to a highly imbalanced dataset, as illustrated in Fig. 2. For example, in iNaturalist dataset 2017 [40] for species classification, there are a total of 5089 classes with the largest classes more than 1000 samples and the smallest classes fewer than 10. In iNaturalist competition 2019, even with an effort to filter out species that have insufficient observations and to further cap the maximum class size to be 500, there is still serious imbalance in the dataset as the smallest classes around 10 samples. Similar data distribution can be observed in other applications, such as a UAV-based object detection dataset [49] and COCO [28].

Like many supervised learning algorithms, the performance of deep neural networks also suffers when the training data is highly imbalanced [9]. The problem can get worse when the categories with fewer data are severely undersampled to the extent that the variation within each category is not fully captured by the given data [41, 2, 44].

The common presence of long-tailed data in real-world problems has led to several effective practices to achieve an overall performance improvement of a given machine learning model. For example, *data manipulation* such as augmen-

tation, under-sampling and over-sampling [4, 14, 10], and *balanced loss function design* (*e.g.*, focal-loss [27] and class-balanced loss [9]), are the two mainstream approaches. These practices often improve the performance reasonably yet the improvement deteriorates when certain categories are severely under-represented, as shown in Fig. 1. Specifically, these methods are often designed to move the class decision boundary to reduce the bias introduced by imbalanced classes. However, when a class is severely under-represented such that it is hard to draw its complete data distribution, finding the right direction to adjust the decision boundary becomes challenging. We therefore focus on exploring the information learned from the head classes (the ones with ample samples) to help the tail classes (the under-represented ones) in a long-tailed dataset.

In this work, we present a novel method to address the long-tailed data classification problem by augmenting the tail classes in the feature space using the information from the head classes. In particular, we insert an attention unit with the help of the class activation map (CAM) [46] to filter out class-specific features and class-generic features from each class. For the samples of each tail class, we augment the high-level features (from high-level deep network layers) by mixing its class-specific features with the class-generic features from the head classes. This method is based on two underlying assumptions: 1) information from the head classes, represented as class-generic features, can help to recover the distribution of tail classes; and 2) the class-generic and class-specific features can be extracted and re-mixed to generate novel samples in the high-level feature space due to a more "linear" representation at that level. We have designed an end-to-end training pipeline to efficiently perform such feature space augmentation, and evaluated our method on artificially created long-tailed CIFAR-10 and CIFAR-100 datasets [24], ImageNet-LT, Places-LT [29] and naturally long-tailed datasets such as iNaturalist 2017 & 2018 [40]. Our approach has shown the state of the art performance on these long-tailed datasets compared to other mainstream deep learning models on data imbalance problems.

2 Related Work

In this section, we first discuss the two directly related approaches, *learning with balanced loss* and *data augmentation*, and then discuss the difference and relation of our approach to few-shot learning and transfer learning.

2.1 Learning with Balanced Loss

One of the most common and often most effective practices is *learning with balanced loss*. The key to such approaches is to counter the effect of a skewed data distribution by adjusting the weights of the samples from the small classes in the loss function. It is typically accomplished by: 1) over-sampling or under-sampling [37, 4, 50, 14, 22] to achieve an even data distribution across various classes, and/or 2) assigning proper weights to the loss terms corresponding to the tail classes [39, 20, 23, 27, 9, 48, 35, 11, 45].

Specifically, these approaches treat the issue of long-tailed data as an optimization problem such that an "optimal" classification boundary can be recovered by carefully adjusting the weight/frequency of each data point in the training set. They typically have the advantage of a relatively clean implementation by either adjusting the loss function [9, 27] or manipulating the input batch [14, 37, 4, 50], and hence were widely adopted in practice [37, 27, 50]. However, when the samples of the tail classes are far from sufficient to recover the true distribution, the performance of such methods deteriorates [9].

Two works along this direction, class-balanced loss [9] and focal loss [27] draw our attention in particular for their generic applications on deep learning models. Specifically, focal loss weights the loss term of each sample based on the probability generated from the last soft-max layer [27]. It implicitly gives higher weights to samples from the tail classes to counter the bias introduced by the sample size. In [9], the concept of the effective number is introduced to calculate the weight of each class in the loss term.

Note that our approach can be used jointly with approaches such as focal loss [27] and potentially gain the benefits from both. For example, we can use the feature space augmentation approach to facilitate the performance of the tail classes, and at the same time, balanced loss methods such as focal loss can give higher weights to the hard examples regardless of the class label during training.

2.2 Data Synthesis and Augmentation

Generating and synthesizing new samples to compensate the small sample size of a tail class is a natural way to improve the performance of deep learning models on long-tailed data. These samples can be either generated from similar samples [5] or synthesized based on the given information of a dataset [18, 50, 15]. The general application of data augmentation in different deep learning models also boosts the interest of developing more sophisticated data augmentation methods. For example, in [7], an input image is partitioned into local regions which are shuffled during training. In [8], local regions of an image are replaced by unlabeled data to generate synthetic images to help training. In [41], a parametric generative model takes noise and existed samples to hallucinate new samples to support training.

As directly manipulating the raw input images may as well introduce unexpected noise, feature vectors are instead generated by training a function to learn the relation between a pair of samples from one class and applies it to the samples in another [17]. Furthermore, recent progress in generative adversarial networks (GAN) have inspired advanced methods using generative models to address the data insufficiency problem [43].

In contrast to the existing approaches on augmenting feature vectors, we focus on modeling the feature space itself rather than training a heavily parametric model that applies to all different classes. The decomposition of feature space is then used to formulate novel training samples in the feature space on the fly.

2.3 Transfer Learning

Past works in the domain of transfer learning and few-shot learning [42, 2, 32, 44, 31, 47] have been conducted to solve the long-tailed problem. Our work shares a similar assumption with these works that the information from the head classes can be used to help the tail classes. However, we explicitly distinguish the generic features and specific features from each class instead of making strong assumptions on the general transferability of knowledge from the head classes to the tail classes. Specifically, in [42], a meta-network is trained to predict the many-shot parameters from few-shot model parameters using data from the head classes with the assumption that the model parameters from different classes share a similar dynamic behavior even if the size of the training set varies. In [2], the representation is shared in general with different embedding approaches across different classes. In [44], the variance of the head classes is learned and transferred to the tail classes with the underlying assumption that each class has its own mean but a shared variance. In [32], visually similar classes are clustered together in order to reduce the level of data imbalance. Knowledge can then be transferred from each cluster to its sub-classes during the fine tuning stage of deep networks for object detection specifically.

In comparison, we intentionally separate the features of each class into class-specific features and class-generic features. Only class-generic features from head classes are seen as transferable knowledge and are hence used for feature space augmentation on the tail classes.

3 The Problem of Long Tail

In this section, we first analyze the underlying issues of long-tailed data that affect model performance (Sec. 3.1), and then explore deeper into the feature space of DNNs and illustrate a novel way to alleviate the problem (Sec. 3.2).

3.1 Two Reasons of Model Performance Drop

Long-tailed data hurt the performance of learning-based classification models mainly due to the following two issues: (1) data imbalance which is relatively easy to solve, and (2) missing coverage of the data distribution caused by limited data, which is harder to deal with.

The data imbalance issue has been discussed in several recent works [9, 4, 14, 27] with good solutions proposed to minimize its impact. This problem is essentially about the bias introduced by the different number of samples in the dataset. With carefully designed sampling schemes and/or loss weights, we can compensate this negative impact and move the classification decision boundary in the right direction. For example, a common practice of training on an imbalanced dataset is to over-sample the small classes or under-sample the large classes [10]. This is built upon the assumption that the underlying decision boundary is indeed well-defined with the given data, and hence with careful adjustment we can find its optimal location.

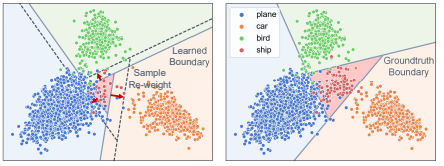

Fig. 3: The difference between the two "optimal" decision boundaries.

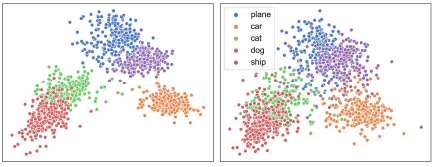

Fig. 4: Left: class-specific features, Right: class-generic features.

However, when there is simply no sufficient data for the tail classes to recover their underlying distribution, the problem of finding an optimal decision boundary becomes ill-defined. In this scenario, it becomes extremely difficult to guess the location of the decision boundary without recovering the distribution first. We hypothesize that the knowledge obtained from the head classes can help with solving the issue.

We further elaborate the issue in Fig. 3 by plotting the feature distribution of 4 classes in CIFAR-10. The features are from the last fully-connected (FC) layer of ResNet-18 and then embedded in 2-D space. When the ship class is under-represented, as shown in the left graph, simply moving the decision boundary will not provide the optimal decision boundary (as shown in the right graph) as if there were sufficient samples.

3.2 Class Activation Map and Feature Decomposition

With limited data in the tail classes and ample data in the head classes, it seems natural to use the knowledge learned from the head classes to help recovering the missing information in the tail classes. However, we have to be careful to differentiate the class-generic information that can be used to recover the distribution of the tail classes from the class-specific information that may mislead the recovery of the distribution of the tail classes.

Inspired by the recent works on attention and visual explanation [46, 36], we find that deep neural network features can be decomposed into two such components in a similar fashion. In particular, let us define class activation map M_c of class c as in [46],

$$M_c(x, y) = \sum_k w_k^c f_k(x, y), \tag{1}$$

where $f_k(x, y)$ is the feature vector in location (x, y) of channel k, and w_k^c the weights of the last layer of classifier corresponding to class c. The larger value of $M_c(x, y)$, the more important of feature vector at (x, y) is to class c, and vice versa.

We further normalize the value of $M_c(x, y)$ to the range of 0 and 1. Therefore, given a pair of thresholds $0 < \tau_s, \tau_g < 1$, we can decompose the class activation

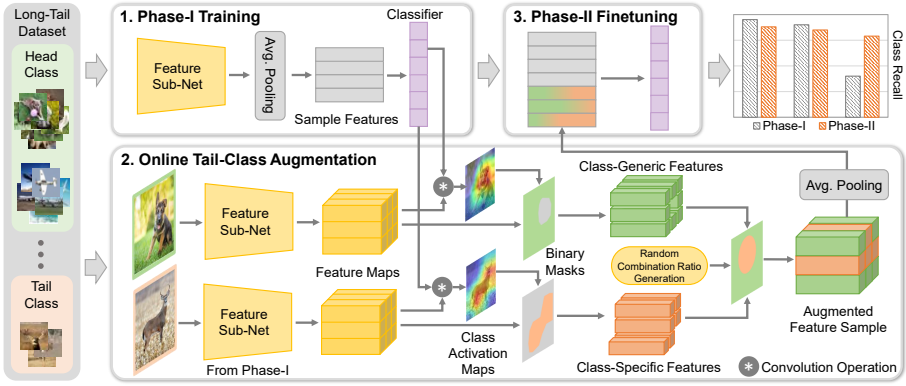

Fig. 5: Overview of the proposed two-phase learning scheme.

map M_c into two parts, M_c^s and M_c^g, to separate the feature vectors into class-specific features and class-generic features as follows,

$$M_c^s = \text{sgn}(M_c - \tau_s) \odot M_c, \tag{2}$$

$$M_c^g = \text{sgn}(\tau_g - M_c) \odot M_c, \tag{3}$$

where $\odot$ is the Hadamard product between two tensors, $\text{sgn}(x) = 1$ for $x \geq 0$ and $\text{sgn}(x) = 0$ for $x < 0$.

Fig. 4 shows the scatter plot of class-generic features and class-specific features of different classes from CIFAR-10 (More results can be seen in the supplemental material). We can see that after decomposition, even when embedded in a 2-D space, the class-specific features are clearly more separated than class-generic features. In general, we have observed a much stronger correlation between class-generic features than class-specific features across different classes and different datasets. These results further substantiate our approach on using class-generic features to augment the tail classes during training.

4 Method

We propose a two-phase training scheme to leverage the class-generic information to recover the distribution of tail classes, as shown in Fig. 5. In Phase-I, samples from all classes are used to learn the feature representation and a base classifier. In Phase-II, online feature space augmentation is applied to generate novel samples for tail classes.

4.1 Initial Feature Learning

In the Phase-I training, we use all images in the dataset to learn the feature sub-network and the base classifier. In order to calculate the class activation maps

in the following steps, we choose a network architecture that contains a single FC layer as the final classifier, which takes input from a global average pooling layer as illustrated in Fig. 5. A number of the modern deep convolutional neural network architectures fit into this category, e.g., ResNet [19], DenseNet [21], MobileNet [34], and EfficientNet [38].

4.2 Feature Space Augmentation

With the pre-trained feature sub-network and the classifier, augmented samples can be generated in the feature space on the fly by mixing the class-specific features from the given tail class and the class-generic features.

One question we need to address is that, given a tail class, how to choose the classes from which the class-generic features will be extracted. A naive solution would be to randomly select the classes from the training dataset. However, the class-generic features from different classes may vary with each other, and such features of a randomly selected class cannot always guarantee a good recovery of the classification decision boundary. From the perspective of the optimal classification decision boundary, we observe that the "nearby" classes in the feature space, i.e. the most "confusing" classes with respect to the given tail class, have the biggest impact on recovering the previously ill-defined decision boundary, as seen in Fig. 3. Specifically, we calculate the classification scores for all other classes for each training sample in a given tail class, and then find its top N_f confusing classes by ranking the average classification scores of other classes over all samples within the tail class.

As described in Sec. 3.2, we use class activation maps to separate the class-generic and class-specific information from a given image. As shown in Fig. 5, the feature sub-network trained in Phase-I is used to extract feature maps for each input image. The weights of the linear classifier trained in Phase-I are adapted to form the 1×1 convolutional filter for each class. For each input image, the filter associated with the ground truth class is applied on its feature maps to generate the class activation map which is further normalized to the range of $[0, 1]$ for consistency. Two independent thresholds τ_g and τ_s are used to extract the corresponding binarized masks for class-generic features and class-specific features following Eq. 2.

The class-generic information in the confusing classes is then leveraged to generate the augmented samples of each tail class in order to recover its intrinsic data distribution. Directly blending information at the pixel level often introduces artificial edges and hence imposes bias to the augmented samples. We, therefore, conduct the fusion in the feature space to suppress the noise and potential bias. In particular, for each real sample in the tail class, we sample N_a images from its N_f confusing classes. The class-specific features from the sample are then combined with the class-generic features from the N_a samples in a linear way. A random combination ratio is generated to guide the fusion by randomly drawing class-generic and class-specific feature vectors to form an augmented sample for the tail class. By randomly modulating the combination ratio between the class-generic and class-specific features, the sample variance

Algorithm 1 Online Feature Augmentation

Input: All training images features $\mathbb{F}$ and their CAM $\mathbb{M}$.
Output: Training batch with augmented feature samples b_{out}.
Initialize output batch b_{out}.
for $i = 1, \ldots, N_t$ **do**
  $\mathbf{F}_c \leftarrow$ Draw one sample from tail classes
  Append $\mathbf{F}_c$ to b_{out}
  $M_c^s \leftarrow M_c > \tau_s$
  $\mathbf{F}_c^s \leftarrow M_c^s \odot \mathbf{F}_c$
  $\{u\} \leftarrow$ Find confusing classes for class c
  for u in $\{u\}$ **do**
    $\mathbf{F}_u \leftarrow$ Draw one sample from class u
    $M_u^g \leftarrow M_u < \tau_g$
    $\mathbf{F}_u^g \leftarrow M_u^g \odot \mathbf{F}_u$
    Generate combination ratio $\gamma \in (0, 1)$
    ▷ Total L spatial locations in $\mathbf{F}_i$
    ▷ Draw with repeat and excluding all zeros feature
    $\{\mathbf{f}_c^s\} \leftarrow$ Draw γL feature vectors from $\mathbf{F}_c^s$
    $\{\mathbf{f}_u^g\} \leftarrow$ Draw $(1-\gamma)L$ feature vectors from $\mathbf{F}_u^g$
    $\mathbf{F}_c^{aug} \leftarrow$ Merge $\{\mathbf{f}_c^s\}$ and $\{\mathbf{f}_u^g\}$
    Append $\mathbf{F}_c^{aug}$ to b_{out}
  end for
  $\{\mathbf{F}_k\} \leftarrow$ Draw $N_t(1 + N_a)$ samples from head classes
  Append $\{\mathbf{F}_k\}$ to b_{out}
end for

is built into this augmentation procedure. In the end, a total of N_a augmented samples are generated for each real sample from the tail class.

4.3 Fine Tuning with Online Augmented Samples

The augmented samples are generated online to fine tune the network trained in Phase-I to improve the performance of the tail classes. In each batch, we sample N_t images from the tail classes, and generate N_a augmented samples online for each of the real samples, which creates a batch including $N_t(1 + N_a)$ samples from the tail classes. The same number of images are also randomly drawn from the head classes to balance the distribution. Thus, a batch of size $2N_t(1 + N_a)$ is generated online for each fine tuning iteration. We summarize this process in Alg. 1.

Fine tuning is performed on the layers after the features being extracted. Since the augmentation is conducted in the feature space, augmented samples can be generated at any stage of the network. However, the deeper features, compared to its shallow counterparts, are more linearly separable, which greatly help the fusion of features from the tail classes and their confusing classes. Moreover, richer spatial information in the lower-level feature maps may introduce artifacts to bias the model training. We analyze the detailed effect of augmenting

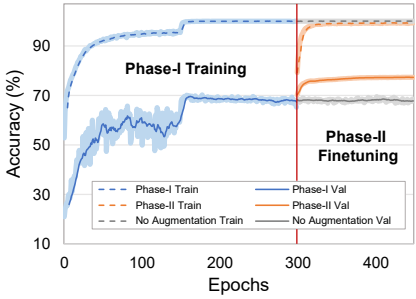

Fig. 6: Learning Curve for long-tailed CIFAR-10 with an imbalance factor of 200 using ResNet-18. The overall accuracy of the validation dataset is illustrated.

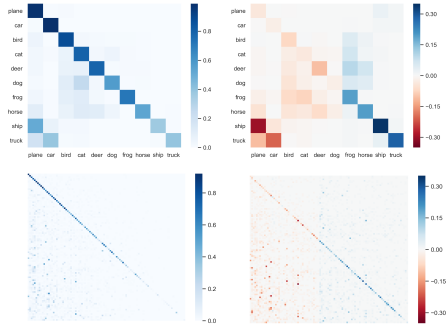

Fig. 7: Confusion matrix for CIFAR-10 (upper) and CIFAR-100 (bottom) at IM 200 using ResNet-18.

samples at different depths in Sec. 5.4. We choose the features right before the last average pooling layer to help with the classification performance and at the same time to realize a simple design. Since the average pooling layer accumulates features in all spatial locations, the spatial distribution of class-generic and class-specific features become irrelevant in the augmented samples. Therefore, when combining, only the ratio between the two types of features needs to be given. Finally, we use the augmented batches to fine tune the FC classifier layer as shown in Fig. 5.

5 Experiments

We conduct experiments on the artificially created long-tailed CIFAR dataset [9] with various simulated imbalance factors, ImageNet-LT [29], Places-LT [29] and the real world long-tailed iNaturalist 2017 and 2018 [40] datasets to validate the proposed method. Deep residual network (ResNet) with various depth are employed in our experiments.

5.1 Long-tailed CIFAR

To demonstrate the effectiveness of the proposed method, long-tailed versions of CIFAR dataset are generated following the protocol mentioned in [9] as $IM = \max(\{N_i\})/\min(\{N_i\})$. Five datasets of different imbalance factors, {10, 20, 50, 100, 200}, are created for both CIFAR-10 and CIFAR-100, where an imbalance factor is defined as where N_i is the number of training samples of the i-th class. ResNet with depth 18 and 34 are adapted for this experiment. We use the original validation set of the CIFAR-10 and CIFAR-100 to evaluate the performance.

The baseline network and the proposed method are implemented in PyTorch and run on a Xeon CPU of 2.1 GHz and Tesla V100 GPU. The initial learning

	ResNet-18					ResNet-34				
IM	10	20	50	100	200	10	20	50	100	200
Baseline	90.73	87.24	82.32	75.16	70.22	91.03	87.32	82.74	78.58	71.42
CB [9] $\beta = 0.9$	90.79	86.61	81.9	75.16	69.16	91.03	87.18	82.48	75.99	70.0
CB [9] $\beta = 0.999$	90.54	86.83	81.81	76.4	69.83	90.74	87.24	81.66	74.85	70.08
CB [9] $\beta = 0.9999$	89.61	86.05	80.4	75.04	69.21	90.69	86.9	81.06	75.74	68.79
FL [27] $\gamma = 0.5$	90.66	86.61	81.55	74.99	69.06	90.76	87.18	81.91	76.5	69.87
FL [27] $\gamma = 1.0$	90.59	86.83	81.79	74.07	68.23	90.7	87.24	81.34	76.44	70.02
FL [27] $\gamma = 2.0$	90.5	86.05	81.25	75.13	68.27	90.08	86.9	82.44	75.58	69.87
SLA [26]	-	-	-	-	-	89.58	-	-	80.24	-
Ours	**91.75**	**88.54**	**84.51**	**80.57**	**77.06**	**91.2**	**89.26**	**84.49**	**82.06**	**75.52**

Table 1: Classification accuracy on long-tailed CIFAR-10.

	ResNet-18					ResNet-34				
IM	10	20	50	100	200	10	20	50	100	200
Baseline	62.59	57.09	48.55	43.65	38.87	63.87	57.55	48.07	43.55	37.5
CB [9] $\beta = 0.9$	63.1	57.02	48.15	43.51	38.58	64.14	58.03	48.44	42.94	38.84
CB [9] $\beta = 0.999$	61.76	55.3	44.28	32.19	26.61	63.05	54.13	40.89	32.65	26.2
CB [9] $\beta = 0.9999$	60.71	53.93	42.02	31.32	25.91	62.28	53.64	40.03	29.82	26.63
FL [27] $\gamma = 0.5$	62.64	57.02	47.9	42.82	38.73	64.36	58.45	48.31	42.72	36.18
FL [27] $\gamma = 1.0$	62.85	57.22	47.76	42.81	40.47	64.83	58.78	48.24	42.64	37.29
FL [27] $\gamma = 2.0$	63.37	57.15	47.0	42.18	40.31	64.48	58.55	47.47	43.33	38.11
SLA [26]	-	-	-	-	-	59.89	-	-	45.53	-
Ours	**65.08**	**58.69**	**51.9**	**46.57**	**42.84**	**65.29**	**59.75**	**52.17**	**48.51**	**41.46**

Table 2: Classification accuracy on long-tailed CIFAR-100.

rate for Phase-I is 0.1 and decreases by 1/10 every 150 epochs. The feature sub-network and base classifier are trained for 300 epochs. In Phase-II, the learning rate is fixed at 0.001. The classifier is fine tuned for 6,400 iterations with a batch size of 128. For each real sample in tail classes, we choose $N_a = N_f = 3$.

A sample learning curve for long-tailed CIFAR-10 with an imbalance factor of 200 using ResNet-18 is shown in Fig. 6. The performance of the proposed method is compared with a baseline setting of the same learning rate but without the feature space augmentation. After the Phase-II feature space augmentation, the accuracy of the proposed method on the validation set increases about 7% during the fine tuning stage, while no noticeable change in accuracy for the baseline setting is observed.

To further illustrate the improvement, the confusion matrix before Phase-II and its changes are shown in Fig. 7. After Phase-I training, tail classes show poor accuracy on the validation set due to insufficient training samples in those classes. Most mis-classified samples fall into the first several head classes, where most training samples belong to, as indicated in the left bottom corner of the confusion matrix. After Phase-II fine tuning, significant improvement is observed for the diagonal elements of the tail classes. The off-diagonal elements decrease accordingly. Although the accuracy of the head classes decrease slightly, due to dramatic improvement in the tail classes, the overall accuracy still increases.

The complete classification performance on different imbalance factors of the two dataset are shown in Tab. 1 and 2. The method using the same ResNet with cross-entropy loss and traditional data augmentation on input images is referred as Baseline in Tab. 1 and 2. In our experiments, we compare our method with

	ImageNet-LT				Places-LT			
	> 100 Many	$\leqslant 100$ & > 20 Medium	< 20 Few	Overall	> 100 Many	$\leqslant 100$ & > 20 Medium	< 20 Few	Overall
Plain Model [29]	40.9	10.7	0.4	20.9	**45.9**	22.4	0.36	27.2
Lifted Loss [31]	35.8	30.4	17.9	30.8	41.1	35.4	24	35.2
FL [27]	36.4	29.9	16	30.5	41.1	34.8	22.4	34.6
Range Loss [45]	35.8	30.3	17.6	30.7	41.1	35.4	23.2	35.1
FSLwF [15]	40.9	22.1	15	28.4	43.9	29.9	**29.5**	34.9
OLTR [29]	43.2	**35.1**	**18.5**	**35.6**	44.7	37	25.3	35.9
Ours	**47.3**	31.6	14.7	35.2	42.8	**37.5**	22.7	**36.4**
Ours+FL	47.0	31.3	16.8	35.3	42.2	36.4	24.0	36.0

Table 3: Top-1 classification accuracy on ImageNet-LT and Places-LT.

the state of the arts on addressing the long-tailed problem, including Class-balanced (CB) loss [9] based method and Focal Loss (FL) from [27] with various choices of hyper-parameters and augmentation based method [26]. Our method outperforms all other methods in both datasets.

5.2 ImageNet-LT and Places-LT Dataset

We also evaluate the proposed method on two constructed large-scale long-tailed datasets ImageNet-LT and Places-LT [29]. ImageNet-LT is a subset of the ILSVRC2012 dataset. Its training set is drawn from the original training set following the Pareto distribution with $\alpha = 6$, which results 115.8K images from 1000 categories with a maximum of 1280 images per category and a minimum of 5 images per category ($IM = 256$). The original validation set with balanced 50K images is used as test set in our experiments. Places-LT dataset is constructed similarly with ImageNet-LT from Places-2 dataset. Finally, 184.5K images from 365 categories are collected, where the largest class contains 4980 images while the smallest ones with 5 images ($IM = 996$). The test set contains balanced 36.5K images.

For fair comparison, we use the same scratch ResNet-10 for ImageNet-LT and pre-trained ResNet-152 for Places-LT as in [29]. The numerical results and comparison with other peer methods are reported in Tab. 3. We also evaluate the combination of other balanced loss methods with the proposed method in these experiments. The different losses are applied in the Phase-I training. We use "Ours+FL" to refer the experiments using Focal Loss and "Ours" for ordinary cross-entropy loss. Both of our methods achieve comparable performance with the state-of-the-art method.

Note that, "Ours+FL" shows better performance than "Ours" in the ImageNet-LT dataset while "Ours" is better in Places-LT. Feature maps generated in the shallow network as ResNet-10 is not as sparse as in ResNet-152. Therefore, as explained in Sec. 4.3, the feature space augmentation delivers more performance boost to ResNet-152. On the other hand, our class balanced training batch generation achieves a similar effect as other balanced loss methods in the Phase-II fine tuning. But applying those losses in the Phase-I may still improve the performance when poor Phase-I performance affects CAM quality.

		iNaturalist 2017	iNaturalist 2018
Baseline	ResNet-50	60.50	62.27
	ResNet-101	61.81	65.19
	ResNet-152	65.12	66.17
CB [9]	ResNet-50*	58.08	61.12
	ResNet-101*	60.94	63.88
	ResNet-152	64.75	66.97
Ours	ResNet-50	**61.96**	**65.91**
	ResNet-101	**64.16**	**68.39**
	ResNet-152	**66.58**	**69.08**

Table 4: Top-1 classification accuracy on iNaturalist (*: results from literature).

5.3 iNaturalist

iNaturalist is a real-world fine-grained species classification dataset. Its 2017 version contains 579,184 training images of 5,089 categories, and its 2018 version [40] has 437,513 training samples in 8,142 classes. The imbalance factor for iNaturalist 2017 is 435 and 500 for iNaturalist 2018. For both versions, there are three validation samples for each class. We adapt ResNet-50, ResNet-101 and ResNet-152 in our experiments, all with 224×224 input image size. The similar training strategy with CIFAR datasets is adapted for iNaturalist. In Phase-I, the starting learning rate is 0.1 and reduced every 30 epochs for total of 100 epochs. In Phase-II, fine tuning is performed with a fixed learning rate of 0.001 for 200 iterations. The top-1 classification accuracy for the validation set of the two datasets are reported in Tab. 4. We compare the proposed method with class-balanced cross-entropy loss on ResNet-152 and class-balanced focal loss on ResNet-101/50. Our method has shown the best performance in all the settings.

5.4 Ablation Analysis

One major hyper-parameter in the proposed method is how to separate the head classes from the tail classes. Specifically, the classes are first sorted in the descent order by the number of training samples in each class as illustrated in Fig. 2. The first h classes are chosen as head classes. In order to unify the choice between different imbalance factors of datasets, we introduce $h_r \in (0, 1)$ which is the ratio between the number of samples in the head classes and the total number of samples. Different h_r choices against the Phase-II classification accuracy are evaluated in Fig. 8. The curves among different datasets show peaks around $h_r = 0.95$. On the left of peaks, fewer samples or classes are used as the head class, and thus class-generic features cannot be drawn sufficiently for feature augmentation. On the right side, fewer classes are selected as the tail classes, and therefore some classes with insufficient training samples will not be fine tuned with augmented samples. For consistency, we choose the minimum of h that satisfies $h_r \geq 0.9$ in all the CIFAR experiments.

We also investigate the classification performance when applying the feature space augmentation at different depths of the network. The ResNet architecture we adapted usually consists of four convolutional blocks. We plug our feature

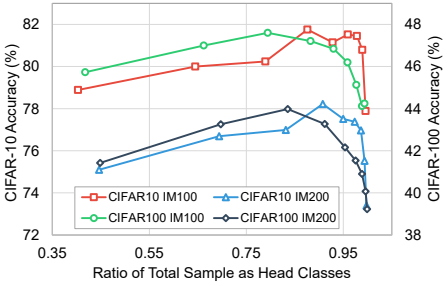

Fig. 8: Phase-II performance dependence on the ratio of total training samples used as the head class sample.

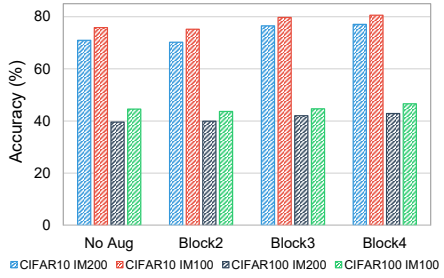

Fig. 9: Classification accuracy by applying feature space augmentation at different depth of ResNet-18.

space augmentation after each of the last three convolutional blocks of ResNet-18 on CIFAR dataset. When augmenting features after Block2 and Block3, class-specific features in $\mathbf{F}_u^g$ are replaced with the class-specific features in $\mathbf{F}_c^g$ with random ratio to generate augmented samples, where spatial information of $\mathbf{F}_u^g$ is preserved. The corresponding classification accuracy after Phase-II fine tuning is shown in Fig. 9. From Fig. 9, one can observe that feature augmentation after Block4 gains the best performance among different datasets and imbalance factors. The feature maps closer to the input side contain more spatial information, which also introduces additional artifacts into the augmented samples. Features generated by Block4 are directly passed into the global pooling layer where the noise in the spatial dimension introduced by augmentation can be eliminated. Moreover, the linearity of the high-level feature space helps the final linear operation of the fusion. We, therefore, apply the feature space augmentation after Block4. We also compare the performance of only sampling balanced finetuning batch without augmentation applied, which is refered as "No Aug" in Fig. 9.

6 Conclusion

In this paper, we propose a novel learning scheme to address the problem of training the deep convolutional neural network based classifier with long-tailed datasets. In detail, by combining the class-generic features in head classes with class-specific features in tail classes, augmented samples are online generated to enhance the performance of the tail classes. Results on long-tailed version CIFAR-10/100, ImageNet-LT, Places-LT and real-world long-tailed iNaturalist 2017/2018 datasets have shown the effectiveness of proposed method.

Acknowledgment Ling was supported in part by US NSF Grants 1814745, 1618398, and 2002434.

References

1. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: CVPR (2015) 2
2. Bengio, S.: Sharing representations for long tail computer vision problems. In: ICMI (2015) 2, 5
3. Bian, X., Lim, S.N., Zhou, N.: Multiscale fully convolutional network with application to industrial inspection. In: WACV (2016) 1
4. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks (2018) 3, 4, 5
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. JAIR (2002) 4
6. Chen, C., Seff, A., Kornhauser, A., Xiao, J.: Deepdriving: Learning affordance for direct perception in autonomous driving. In: ICCV (2015) 1
7. Chen, Y., Bai, Y., Zhang, W., Mei, T.: Destruction and construction learning for fine-grained image recognition. In: CVPR (2019) 4
8. Chen, Z., Fu, Y., Chen, K., Jiang, Y.G.: Image block augmentation for one-shot learning. In: AAAI (2019) 4
9. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR (2019) 2, 3, 4, 5, 10, 11, 12, 13
10. Drummond, C., Holte, R.C., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on learning from imbalanced datasets II (2003) 3, 5
11. Elkan, C.: The foundations of cost-sensitive learning (2001) 3
12. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J.: A guide to deep learning in healthcare. Nature medicine (2019) 1
13. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010) 2
14. Geifman, Y., El-Yaniv, R.: Deep active learning over the long tail. arXiv:1711.00941 (2017) 3, 4, 5
15. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: CVPR (2018) 4, 12
16. Girshick, R.: Fast r-cnn. In: ICCV (2015) 1
17. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: CVPR (2017) 4
18. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (2008) 4
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 1, 8
20. Huang, C., Li, Y., Change Loy, C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR (2016) 3
21. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017) 8
22. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: ICLR (2020) 3
23. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. TNNLS (2017) 3

24. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009) 2, 3
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012) 1
26. Lee, H., Hwang, S.J., Shin, J.: Rethinking data augmentation: Self-supervision and self-distillation. arXiv preprint arXiv:1910.05872 (2019) 11, 12
27. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) 3, 4, 5, 11, 12
28. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 2
29. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR (2019) 3, 10, 12
30. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015) 1
31. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: CVPR (2016) 5, 12
32. Ouyang, W., Wang, X., Zhang, C., Yang, X.: Factors in finetuning deep model for object detection with long-tail distribution. In: CVPR (2016) 5
33. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (2015) 2
34. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: CVPR (2018) 8
35. Sarafianos, N., Xu, X., Kakadiaris, I.A.: Deep imbalanced attribute classification using visual attention aggregation. In: ECCV (2018) 3
36. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017) 6
37. Shen, L., Lin, Z., Huang, Q.: Relay backpropagation for effective learning of deep convolutional neural networks. In: ECCV (2016) 3, 4
38. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019) 8
39. Ting, K.M.: A comparative study of cost-sensitive boosting algorithms. In: ICML (2000) 3
40. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: CVPR (2018) 2, 3, 10, 13
41. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: CVPR (2018) 2, 4
42. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. In: NIPS (2017) 5
43. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: CVPR (2018) 4
44. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Feature transfer learning for deep face recognition with long-tail data. arXiv:1803.09014 (2018) 2, 5
45. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: ICCV (2017) 3, 12
46. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016) 3, 6
47. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR (2020) 5

48. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. TKDE (2005) 3
49. Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., Wu, H., Nie, Q., Cheng, H., Liu, C., et al.: Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In: ECCV (2018) 1, 2
50. Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV (2018) 3, 4