International Journal of Approximate Reasoning 150 (2022) 83–97

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Hybrid ResNet based on joint basic and attention modules for long-tailed classification

Wei Zhao [a,b], Yuling Su [a,b], Minjie Hu [a,b], Hong Zhao [a,b,*]

[a] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China
[b] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian, 363000, China

ARTICLE INFO

Article history:
Received 15 December 2021
Received in revised form 3 May 2022
Accepted 9 August 2022
Available online 17 August 2022

Keywords:
Deep learning
Long-tailed distribution learning
Hybrid ResNet
Attention module

ABSTRACT

Long-tailed distribution learning is one of the critical research fields of deep learning and has gradually become a research hotspot. Existing re-sampling methods for long-tailed data classification attempt to adjust the number of tail class samples to balance the overall feature space and achieve satisfactory results. However, the methods impair the representative ability of the learned features to a certain extent, which in turn affects the tail class feature space. In this paper, we propose a hybrid ResNet based on joint basic and attention modules to enhance the tail class feature space, which provides rich discriminative and representative features in the tail class feature space. Firstly, we use hybrid ResNet to extract features, where the basic module ResNet and the attention module ResNet extract head and tail class features, respectively. The enhancement of tail class features can reduce the dependence of the classifier on head class features. Secondly, we build a fusion loss function, which considers the tradeoff between head loss and tail loss for long-tailed distribution learning. Experimental results show that the proposed model outperforms several state-of-the-art models in the long-tailed classification. Our model was 2.67% better than the optimal method under the long-tailed Tiny-Imagenet-LT dataset with an imbalanced ratio of 100.

1. Introduction

Long-tailed distribution learning [1,2] has important significance and challenges in the field of deep learning [3,4]. Real-world data usually have long-tailed label distribution [5,6] (as shown in Fig. 1). The characteristics of long-tailed distribution data are that a few class (head class) samples account for the majority, while most class (tail class) samples account for the minority [7]. The imbalanced distribution of data between head and tail classes distorts the overall feature space, i.e., many samples in the head class generate a large feature space, while only a small number of samples in the tail class correspond to a small feature space [8]. This skewed long-tailed data distribution makes training neural network-based classification models challenging. In recent years, existing long-tailed distribution learning models have been widely applied to various research fields, such as face recognition [9,10], species classification [11], and medical image diagnosis [12].

Re-sampling [1,13] is popular for dealing with long-tailed problems. It aims to adjust the samples of different classes to balance the number of samples of head and tail classes. The re-sampling method divides into two strategies, including over-sampling and under-sampling. The first strategy is over-sampling, which copies a few tail samples to balance the

* Corresponding author at: School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China.
E-mail address: hongzhaocn@163.com (H. Zhao).

https://doi.org/10.1016/j.ijar.2022.08.007
0888-613X/

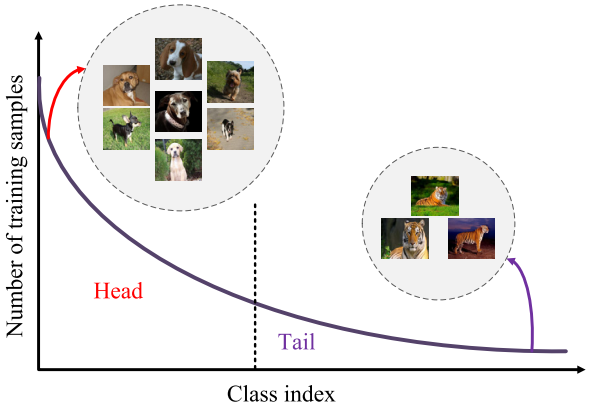

Fig. 1. Structural representation of the long-tailed VOC dataset.

number of samples. Hock et al. [14] demonstrated the effectiveness of the over-sampling technique by comparing the actual performance of whether or not the over-sampled pixel array was used. Chawla et al. [15] proposed to randomly select a part of samples for each tail class within the decision boundary of feature space to synthesize new tail samples. To avoid over-fitting, Han et al. [16] designed a boundary synthesis over-sampling technology by only over-sampling the tail samples near the decision boundary. The tail class feature information is not sufficiently rich by the above over-sampling methods. Barua et al. [17] generated the synthetic samples from the weighted informative minority class samples using a clustering approach. In addition, Liang et al. [18] presented to make the newly generated samples closer to the sample center, which avoids the generation of outlier samples or the change in the distribution of the dataset.

The second strategy is under-sampling, which removes samples from the majority class to adjust the imbalance. For instance, Yen et al. [19] suggested an under-sampling method based on clustering to select representative data as training data. To avoid the loss of head class information, Kim et al. [20] designed a synthetic sampling method to generate tail classes from head classes. This method makes the best of the rich information of head samples without overusing the tail samples. Subsequently, Ahn et al. [21] presented an under-sampling algorithm based on the membership probability of most classes to solve the class imbalance problem with low information loss. Similarly, Rekha et al. [22] proposed an under-sampling method based on critical sample removal, which can minimize the excessive elimination of data and thus reduce the loss of information.

The above-mentioned approaches use the re-sampling methods to tackle the challenges of long-tailed classification and have achieved effective results. However, these methods impair the representative ability of the learned features to a certain extent, which in turn affects the tail class feature space. For instance, the over-sampling method has the risk of overfitting the tail data and cannot provide rich discriminative features to the tail class feature space. The under-sampling method has the risk of missing head data information, and the tail feature space is not enhanced.

Recently, the attention mechanism has achieved great success in the visual domain, which mainly prioritizes task-relevant information and attenuates irrelevant information [23,24]. More specifically, the attention module can make the features extracted by the network focus on the distinguishable features related to the class and reduce the redundant common features such as the background. Therefore, we attempt to embed an attention module into long-tailed data classification, which aims to enhance the tail feature space with an attention module to improve the representativeness of the learned features. Our method makes the overall feature space relatively balanced to improve long-tailed classification performance.

In this paper, we propose a hybrid ResNet based on joint basic and attention modules to research the effect of the attention mechanism for long-tailed learning. Our model consists of two module ResNets, including the basic module ResNet and the attention module ResNet (i.e., BM-ResNet and AM-ResNet). Firstly, we separate the original long-tailed data into head and tail data by experimentally obtaining a threshold. We use hybrid ResNet to extract features for head and tail data. The basic module ResNet extracts head class features, and the attention module ResNet extracts tail class features. The model enhances the tail class features, capturing classification-related features and suppressing background features. Secondly, we utilize head and tail class features to construct head and tail classifiers to obtain the probability of head and tail classes and build a fusion loss function of head loss and tail loss. We consider the impact of the fusion loss function for long-tailed classification. Unlike the traditional long-tailed distribution learning, the model makes full use of the feature of head and tail data and does not impair the representative ability of the learned features.

Experimental results under three long-tailed datasets show that our model is better than the most advanced classification methods. Especially for long-tailed CIFAR-100-LT with an imbalanced ratio of 200 (an extreme imbalance case), our model achieves 40.64% classification accuracy, which is 1.95% better than LDAM-DCB. Similarly, our model achieves 30.1% classification accuracy, which is 2.32% better than the optimal method for long-tailed the Tiny-Imagenet-LT dataset with an imbalanced ratio of 200.

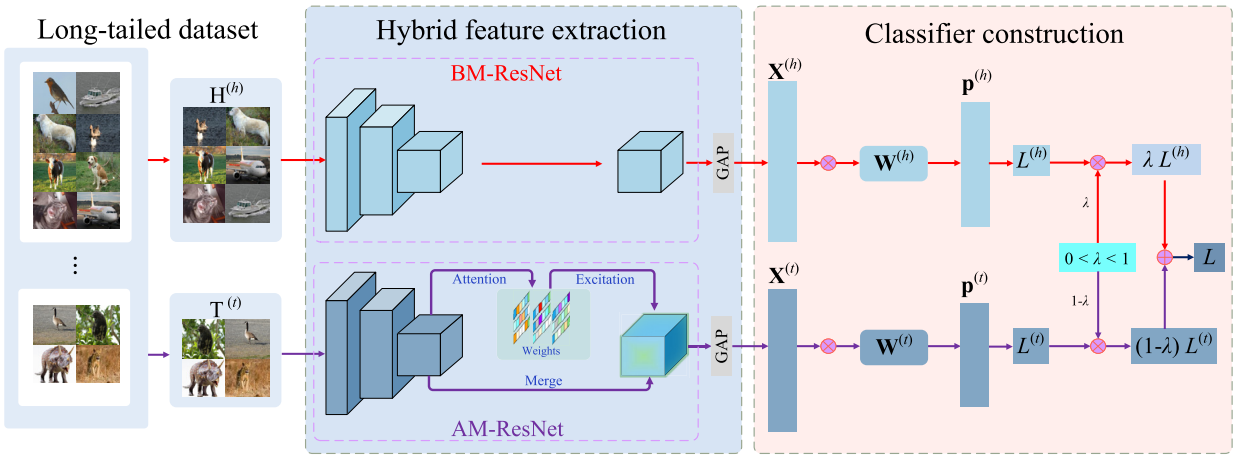

Fig. 2. Framework of the HRJBA model.

The remainder of this paper is organized as follows. In Section 2, we describe the details of our proposed approach. We present the experimental settings used to test our approach in Section 3. The experimental results and analysis are summarized in Section 4, then the paper concludes in Section 5.

2. Hybrid ResNet based on joint basic and attention modules for long-tailed classification

In this section, we introduce the main framework of the hybrid ResNet based on joint basic and attention modules for the long-tailed classification (HRJBA) model.

2.1. Basic framework

The basic flowchart of the HRJBA model is shown in Fig. 2. The specific process of the model is as follows: First, the original data is separated into head data and tail data by threshold. Then, the head and tail data are respectively entered into different networks for feature extraction to obtain feature matrices $\mathbf{X}^{(h)}$ and $\mathbf{X}^{(t)}$. The feature matrices are sent to the classifiers $\mathbf{W}^{(h)}$ and $\mathbf{W}^{(t)}$, respectively. Next, we obtain the probabilities of the head and tail classes to calculate the head and tail class losses, respectively. Finally, the fusion of head and tail losses aims to control the influence of head class learning on tail learning by the λ parameter. The fusion loss performs parameter updates through backpropagation.

2.2. Feature extraction based on hybrid ResNet with basic and attention modules

Feature extraction is an indispensable processing step of classification [25–27]. Feature quality [28,29] has a significant influence on a model generalization ability. In long-tailed datasets, we enhance feature extraction for tail classes and ultimately affect the classification effect [30].

We consider allocating the long-tailed dataset to the sample sets of head and tail classes. We define total sample set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ of K classes, where N is the total number of samples and $y \in \{1, \ldots, K\}$ denote the corresponding class label. To better describe our approach, let $N = \sum_{k=1}^{K} N_k$ denote total sample size of $\mathcal{D}$, where N_k is number of the k^{th} class. Let $\mathcal{S} = \{N_1, N_2, \ldots, N_k, \ldots, N_K\}$ denote set of sample numbers of classes. We assume that these classes are ordered by sample numbers descending order, where $N_1 \geq N_2 \geq \ldots \geq N_k \geq \ldots \geq N_K$. We define a threshold ϑ, where the threshold is decimal and will be obtained in the experiment. We obtain the number of labels $b = \vartheta \times K$ for the tail class. Similarly, the number of labels for the head class is $K - b$. Then, we can obtain the boundary value $N_{(K-b)}$, where $N_{(K-b)} \in \mathcal{S}$. The formula for judging that k belongs to the label set of the head classes or the label set of the tail classes is expressed as:

$$k \in \begin{cases} Y^{(h)}, & \text{if } N_k \geqslant N_{(K-b)}; \\ Y^{(t)}, & \text{if } N_k < N_{(K-b)}; \end{cases} \tag{1}$$

where $Y^{(h)}$ is the label set of the head classes and $Y^{(t)}$ is the label set of the tail classes. Through the above operations, we obtain the sample sets of head and tail classes through $Y^{(h)}$ and $Y^{(t)}$ label sets, which can be expressed as:

$$(x_i, y_i) \in \begin{cases} \mathcal{H}^{(h)}, & \text{if } y_i \in Y^{(h)}; \\ \mathcal{T}^{(t)}, & \text{if } y_i \in Y^{(t)}; \end{cases} \tag{2}$$

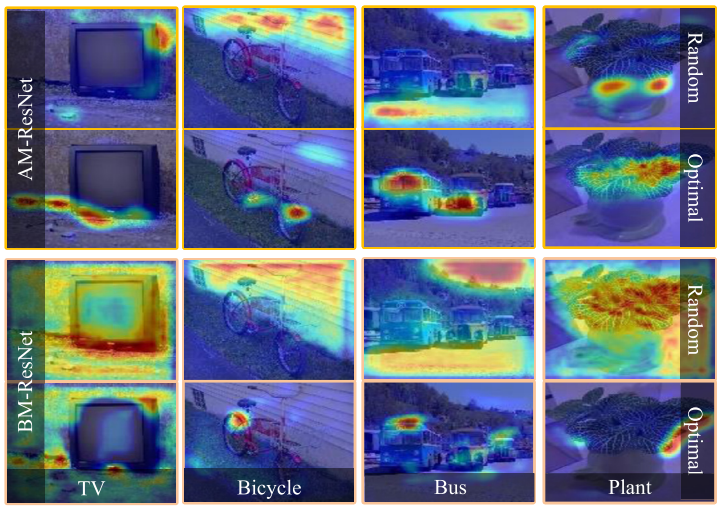

Fig. 3. Visualization of feature activations obtained by different ResNets. *Random* means that BM-ResNet and AM-ResNet load network parameters are randomly in the test phase. *Optimal* means that BM-ResNet and AM-ResNet are trained on the same dataset with consistent settings, and the optimal parameters are obtained. Both networks are loaded with random and optimal parameters in the test phase. The features are extracted on the test set and shown by Grad-CAM.

where $i \in \{1, \ldots, N\}$ and (x_i, y_i) belongs to a sample of $\mathcal{D}$. The total sample set is divided into two sample sets of head and tail classes, which can be expressed as:

$$\mathcal{D} = \mathcal{H}^{(h)} \cup \mathcal{T}^{(t)}, \tag{3}$$

where $\mathcal{H}^{(h)}$ is the sample set of head classes and $\mathcal{T}^{(t)}$ is the sample set of tail classes. Ultimately, we can obtain that $\mathcal{H}^{(h)}$ is the sample set of head classes, and $\mathcal{T}^{(t)}$ is the sample set of tail classes.

Let $\mathbf{X}^{(h)} = [\mathbf{X}_1^{(h)}; \mathbf{X}_2^{(h)}; \ldots; \mathbf{X}_u^{(h)}; \ldots; \mathbf{X}_m^{(h)}]$ be the feature matrix of head classes, where m is the sample numbers of head classes and $\mathbf{X}_u$ is feature vector of a sample. Similarly, let $\mathbf{X}^{(t)} = [\mathbf{X}_1^{(t)}; \mathbf{X}_2^{(t)}; \ldots; \mathbf{X}_v^{(t)}; \ldots; \mathbf{X}_n^{(t)}]$ be the feature matrix of tail classes, where n is the sample numbers of tail classes and $\mathbf{X}_v$ is feature vector of a sample. Subsequently, we assume the basic module feature extraction ResNet as f_ϕ and the attention module feature extraction ResNet as f_ψ. Let (x_u, y_u) and (x_v, y_v) be the samples from $\mathcal{H}^{(h)}$ and $\mathcal{T}^{(t)}$ set. We feed x_u into the f_ϕ and x_v into the f_ψ, which can be obtained as follows:

$$\begin{cases} \mathbf{X}_u^{(h)} = f_\phi(x_u); \\ \mathbf{X}_v^{(t)} = f_\psi(x_v); \end{cases} \tag{4}$$

where $\mathbf{X}_u^{(h)}$ is a feature vector in $\mathbf{X}^{(h)}$ and $\mathbf{X}_v^{(t)}$ is a feature vector in $\mathbf{X}^{(t)}$.

The f_ψ can enhance tail class features, which compare with the f_ϕ. For instance, we choose samples of tail classes and give the feature visualization of the basic module ResNet (i.e., BM-ResNet) and the attention module ResNet (i.e., AM-ResNet) in Fig. 3. We use Grad-CAM [31] to show the attention range of the sample under different networks. From the figure, we can get the following observations: (1) In the case of random parameters, AM-ResNet extracts more specific features than BM-ResNet. On the other hand, BM-ResNet extracts more cluttered features and is not easy to classify. (2) In the case of optimal parameters, AM-ResNet focuses on some primary regions close to the image labels.

2.3. Classifier construction based on hybrid ResNet with basic and attention modules

In this section, we use the head and tail class features to construct head and tail classifiers, respectively, which obtain the fusion loss function. We consider the impact of the fusion loss function for long-tailed classification. In the previous section, we obtain the head class features $\mathbf{X}^{(h)}$ and tail class features $\mathbf{X}^{(t)}$ and take the feature vector $\mathbf{X}_u^{(h)}$ and $\mathbf{X}_v^{(t)}$ of one sample, respectively. Firstly, we construct the head class classifier. The feature vector $\mathbf{X}_u^{(h)}$ is sent into the classifier $\mathbf{W}^{(h)}$ to get the output logits. The output logits are formulated as:

$$\mathbf{Z}^{(h)} = \mathbf{X}_u^{(h)} (\mathbf{W}^{(h)})^\top + \mathbf{b}^{(h)}, \tag{5}$$

where $\mathbf{W}^{(h)}$ is a full connection layer weight matrix in the head class classifier, and $\mathbf{b}^{(h)}$ is a bias vector. The $\mathbf{Z}^{(h)}$ is the predicted out. We obtain the predicted out as $\mathbf{Z}^{(h)} = [z_1^{(h)}, z_2^{(h)}, \ldots, z_m^{(h)}]^\top$, where m is the number of head classes. For each class $u \in \{1, 2, \ldots, m\}$, the softmax function calculates the probability of the class by

$$p_u = \frac{e^{z_u}}{\sum_{j=1}^{m} e^{z_j}}. \tag{6}$$

The output probability vector as $\mathbf{p}^{(h)} = [p_1^{(h)}, p_2^{(h)}, \ldots, p_m^{(h)}]^\top$, where $p_u^{(h)} \in [0, 1]$.

Secondly, we construct the tail class classifier. The feature vector $\mathbf{X}_v^{(t)}$ is sent into the classifier $\mathbf{W}^{(t)}$ to get the output logits. The output logits are formulated as:

$$\mathbf{Z}^{(t)} = \mathbf{X}_v^{(t)}(\mathbf{W}^{(t)})^\top + \mathbf{b}^{(t)}, \tag{7}$$

where $\mathbf{W}^{(t)}$ is a full connection layer weight matrix in the tail class classifier, and $\mathbf{b}^{(t)}$ is a bias vector. The $\mathbf{Z}^{(t)}$ is the predicted out. We obtain the predicted out as $\mathbf{Z}^{(t)} = [z_1^{(t)}, z_1^{(t)}, \ldots, z_n^{(t)}]^\top$, where n is the number of tail classes. For each class $v \in \{1, 2, \ldots, n\}$, the softmax function regards each class as mutual exclusive and calculates the probability distribution over tail classes by

$$p_v = \frac{e^{z_v}}{\sum_{j=1}^{n} e^{z_j}}. \tag{8}$$

The output probability vector as $\mathbf{p}^{(t)} = [p_1^{(t)}, p_2^{(t)}, \ldots, p_n^{(t)}]^\top$, where $p_v^{(t)} \in [0, 1]$.

Through the above description, K is the total number of head and tail classes, i.e., $K = m + n$. Let $\mathcal{L}_{CE} = -\log(p)$ be softmax the cross-entropy loss function, where p indicate the softmax probability of the sample. The loss for samples of head and tail classes is written as:

$$\begin{cases} \mathcal{L}_{CE}^{(h)} = -\log(p_u); \\ \mathcal{L}_{CE}^{(t)} = -\log(p_v); \end{cases} \tag{9}$$

where $\mathcal{L}_{CE}^{(h)}$ is the classification loss of head class and p_u is the probability of the head class sample u. $\mathcal{L}_{CE}^{(t)}$ is the classification loss of the tail class, and p_v is the probability of the tail class sample v. In addition, we propose a fusion loss function to balance head and tail losses, which is expressed as:

$$\mathcal{L} = \lambda \mathcal{L}_{CE}^{(h)} + (1 - \lambda)\mathcal{L}_{CE}^{(t)}, \tag{10}$$

where λ is a positive factor for the fusion loss function. The parameter λ controls the influence of the head class learning on the tail class learning, while the model pays attention to the tail class learning as parameter $1-\lambda$. Note that the parameter λ ranges from 0 to 1. We use the loss function $\mathcal{L}$ to control the two parameters in a more reasonable range and converge our loss function. The whole model is end-to-end trainable.

Algorithm 1 provides the pseudo code for the model training process. We first construct the head class and the tail class sample sets by the threshold and extract the features of the head class and the tail class sample sets, respectively, in lines 4-5. The main training process is listed in lines 6-11. In particular, we calculate the corresponding losses based on the predicted probabilities of the head class samples in lines 6-7. Similarly, we calculate the predicted probabilities of tail class samples and compute the corresponding losses in lines 8-9. Then, we perform the fusion of head class loss and tail class loss in line 10. Finally, we update parameters ϕ, ψ, $\mathbf{W}^{(h)}$, and $\mathbf{W}^{(t)}$ by loss $\mathcal{L}$ back propagation in line 11.

Algorithm 1 Hybrid ResNet based on joint basic and attention modules for long-tailed classification (HRJBA).

Input: Training set $\mathcal{D} = \{(x_1, y_1), ..., (x_i, y_i), ..., (x_N, y_N)\}$, where $y_i \in \{1, 2, ..., K\}$, and N, K represent the number of samples and classes in the training set, respectively. The number of total training epochs is E_{max}. The batch of each epoch is B_{max}. The feature extraction modules are f_ϕ and f_ψ. The classifier parameters for the head and tail classes are $\mathbf{W}^{(h)}$ and $\mathbf{W}^{(t)}$, respectively.
Output: The parameters ϕ, ψ, $\mathbf{W}^{(h)}$, and $\mathbf{W}^{(t)}$.
1: Initialize parameters ϕ, ψ, $\mathbf{W}^{(h)}$, and $\mathbf{W}^{(t)}$;
2: **for** $epoch = 1 : E_{max}$ **do**
3: **for** $batch = 1 : B_{max}$ **do**
4: Select the samples assignment head sample set $\mathcal{H}^{(h)}$ or tail sample set $\mathcal{T}^{(t)}$ according to Eq. (2);
5: Extract the sample features of $\mathcal{H}^{(h)}$ and the sample features of $\mathcal{T}^{(t)}$ by f_ϕ and f_ψ respectively according to Eq. (4);
6: Compute head class probabilities $\mathbf{p}^{(h)}$ based on the head class classifier $\mathbf{W}^{(h)}$ according to Eqs. (5) and (6);
7: Obtain loss of head classes $\mathcal{L}^{(h)}$ according to Eq. (9);
8: Compute head class probabilities $\mathbf{p}^{(t)}$ based on the head class classifier $\mathbf{W}^{(t)}$ according to Eqs. (7) and (8);
9: Obtain loss of tail classes $\mathcal{L}^{(t)}$ according to Eq. (9);
10: Obtain the loss $\mathcal{L}$ by fusing the head and tail losses according to Eq. (10);
11: Update parameters ϕ, ψ, $\mathbf{W}^{(h)}$, and $\mathbf{W}^{(t)}$ by loss $\mathcal{L}$ back propagation;
12: **end for**
13: **end for**

We use Example 1 to give an intuitive explanation of the test process shown in Fig. 4.

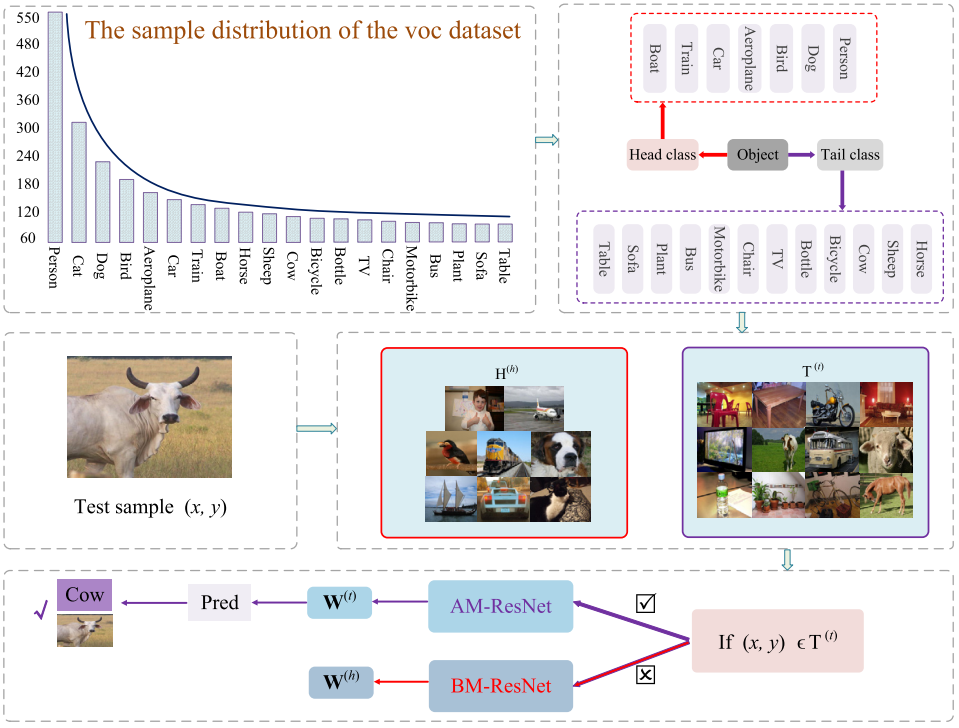

Fig. 4. An example representing the test process of our model.

Example 1. The VOC dataset exhibits a long-tailed distribution. We select a test sample to simulate the classification process of the framework.

(1) We divide the training labels into label sets of head and tail classes by a threshold. Subsequently, the sample sets of head and tail classes are formed by mapping the label sets to the test set.

(2) We choose a test sample. The test sample enters the already trained AM-ResNet network and passes the classifier to generate a prediction if the sample belongs to the tail class set. The final sample predicted label is Cow.

3. Experimental settings

In this section, we introduce the experimental settings from the following aspects: (1) the experimental datasets; (2) the implementation details; (3) the compared methods.

3.1. Dataset descriptions

In our experiments, we use three long-tailed datasets, including VOC-LT, CIFAR-100-LT, and Tiny-Imagenet-LT. We describe the three datasets as follows:

VOC-LT. VOC [32] is a benchmark visual object classification and recognition dataset from the challenge of PASCAL visual object classes. VOC-LT is sampled from the 2012 train-val set of VOC. Thus, the VOC-LT contains 20 class labels in 6976 samples, 3437 samples for training, and 3539 samples for test. The number of samples per class in the training set ranged from 8 to 456, and the number of samples per class in the test set ranged from 13 to 546. The obtained VOC-LT has the characteristics of long-tailed data distribution. Fig. 5(a) shows a histogram of the number of training samples per class for the VOC-LT.

CIFAR-100-LT. The original version of CIFAR-100 [33] contains 50,000 training images and 10,000 test images of size 32 $\times$ 32 with 100 classes. Similarly, the long-tailed dataset contains the same classes. Therefore, to make a fair comparison, we use the same version of the CIFAR dataset as in [1], making the degree of data imbalance controllable. The imbalance factor ρ of the long-tailed CIFAR dataset is defined as the ratio of the number of training samples in the largest class to the number of training samples in the smallest class, i.e., $\rho = \mathrm{N}_{max}/\mathrm{N}_{min}$. The imbalance factor in common standard experiments is 50, 100, and 200. Fig. 5(b) shows a histogram of the number of training samples per class for the CIFAR-100-LT.

Tiny-Imagenet-LT. As the original dataset, Tiny-Imagenet [34,35] contains 200 classes, with a total of 110,000 image samples, 100,000 image samples for training and 10,000 image samples for test. Each class has 500 training image samples and 50 test image samples, 64 $\times$ 64 in size. We use the same strategy as CIFAR-100-LT to create Tiny-Imagenet long-tailed

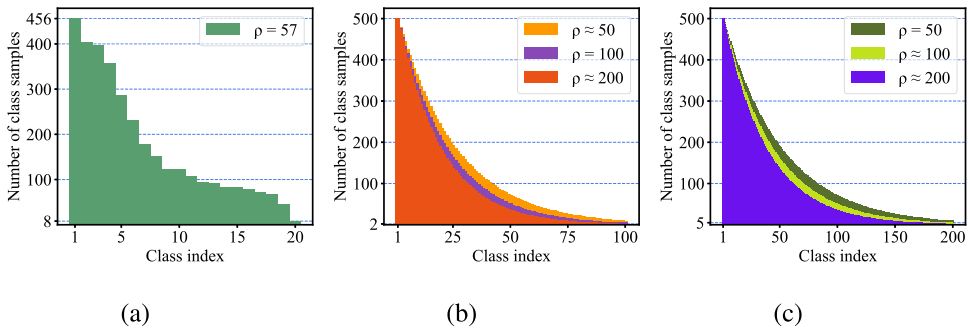

Fig. 5. Histogram of training samples number per class on different datasets. (a) VOC-LT; (b) CIFAR-100-LT; (c) Tiny-Imagenet-LT.

Table 1
Descriptions of the experimental datasets.

Dataset	Imbalance ratio	Training	Test	Label	Image size	Type
VOC-LT	57	3437	3539	20	128	Image
CIFAR-100-LT	50	12067	10000	100	32	Image
CIFAR-100-LT	100	10847	10000	100	32	Image
CIFAR-100-LT	200	9502	10000	100	32	Image
Tiny-Imagenet-LT	50	25082	10000	200	64	Image
Tiny-Imagenet-LT	100	21543	10000	200	64	Image
Tiny-Imagenet-LT	200	18836	10000	200	64	Image

distribution data. Similarly, the imbalance factor ρ of the Tiny-Imagenet dataset is the same as in CIFAR-100-LT, i.e., 50, 100, and 200. Fig. 5(c) shows a histogram of the number of training samples per class for the Tiny-Imagenet-LT.

3.2. Implementation details

We describe the implementation details of three long-tailed datasets. Table 1 shows a detailed description of the experimental dataset.

Implementation details on VOC-LT. For the VOC-LT dataset, each image sample is different in size. All sample image pixels are larger than 300pt. To better represent the image feature information, we perform data augmentation without losing the feature information of the images. Therefore, we use a relatively good data augmentation strategy: Data augmentation strategy for the training set, we firstly resize the image to 128 $\times$ 128 pixels, then 4 pixels are padded on each side, and a 128 $\times$ 128 crops are randomly sampled from the padded image or its horizontal flip.

Implementation details on CIFAR-100-LT. For the CIFAR100-LT dataset, we follow the simple data augmentation in [36] for training: 4 pixels are padded on each side, and a 32 $\times$ 32 crop is randomly sampled from the padded image or its horizontal flip.

Implementation details on Tiny-Imagenet-LT. For the Tiny-Imagenet-LT dataset, we follow the simple data augmentation for training: 4 pixels are padded on each side, and a 64 $\times$ 64 crop is randomly sampled from the padded image or its horizontal flip.

Network selection: (1) For all comparative experiments, we use ResNet32 [37] as backbone network. (2) For our model experiment, we train to use our proposed hybrid ResNet (i.e., BM-ResNet and AM-ResNet, where BM-ResNet is the same as ResNet32). Specific experiment settings: Both the comparative experiment and our experiment use a standard small-batch stochastic gradient descent [38] (i.e., SGD) with a momentum of 0.9 and a weight decay of 2e $\times$ 10^{4}. All the input images are normalized. They are trained with a batch size of 128 for 200 epochs, and the initial learning rate sets to 0.1. Then, we use the linear warm-up learning rate plan [38] to train the first five epochs, and the learning rate attenuated by 0.01 at 160 and 180 epochs, respectively. We train all models on a Ubuntu20.04 desktop computer using NVIDIA GTX3090 with 24.0 GB video memory and a 2.40 GHz $\times$ 24 Intel Xeon Silver 4214R CPU.

3.3. Comparison methods

In this section, we consider a wide range of baseline methods, i.e., a combination of loss functions and strategies. As a comparison experiment, it is detailed as follows:

Loss functions: (a) cross-entropy loss (CE): all samples have the same weight, and standard training perform in CE Loss; (b) focal loss (Focal) [39]: reshaping the standard cross-entropy loss such that it down-weights the loss assigned to well-classified examples; (c) label-distribution-aware margin loss (LDAM) [40]: a label-distribution-aware loss function to encourage larger margins for minority classes.

Table 2
The error rate of each class on the VOC-LT dataset (%).

No.	Class	Sample number	Error rate
1	Person	546	28.21
2	Cat	422	41.47
3	Dog	380	52.89
4	Bird	341	41.06
5	Aeroplane	267	20.22
6	Car	216	34.26
7	Train	195	30.77
8	Boat	138	31.88
9	Horse	130	73.85
10	Sheep	130	53.85
11	Cow	123	69.11
12	Bicycle	98	43.88
13	Bottle	98	86.73
14	TV	95	44.21
15	Chair	85	71.76
16	Motorbike	77	58.44
17	Bus	71	25.35
18	Plant	66	75.76
19	Sofa	48	64.58
20	Table	13	100.00

Table 3
The error rate corresponding to the number of classes on the VOC-LT dataset (%).

No.	Class 1-5	Class 6-10	Class 11-15	Class 16-20
Error rate	37.01	42.52	63.33	57.09

Strategies: (a) re-sampling (RS) [41]: the sampling probability of each sample is the reciprocal of the total samples of the class. Usually used for under-sampling of head samples and over-sampling of tail samples; (b) re-weighting (RW) [42]: the reciprocal of each class total sample is used as the weight in the loss function to balance the sample loss; (c) class-balanced re-weighting (CB-RW) [43]: (RW variant), instead of inverse class frequencies, the samples are re-weighted based inverse of effective number for each class, defined as $(1 - \beta^{N_k}) / (1 - \beta)$. Here, we use $\beta = 0.9999$; (d) deferred re-sampling (DRS) [40]: RS is deferred until the later stage of training of a model; (e) deferred re-weighting (DRW) [40]: RW is deferred until the later stage of training of a model; (f) deferred class-balanced re-weighting (DCB) [43]: CB is deferred until the later stage of training of a model.

When a loss function combines with a strategy, we concatenate them with dashes as a new algorithm. Then, according to the experimental settings in the original paper, we rerun the combinations of these algorithms and take the resulting results as our comparison experiments.

4. Experimental results and analysis

In this section, we introduce and discuss the experimental results from five aspects. In Section 4.1, we analyze the factors that affect the classification accuracy of long-tailed data. In Section 4.2, we compare the effectiveness comparison of BM-ResNet and AM-ResNet. In Section 4.3, we analyze the sensitivity of parameter λ on the fusion function of head and tail losses. In Section 4.4, we compare the performance of our method in local accuracy. In Section 4.5, we use heatmaps as a case to validate our motivation and demonstrate the effectiveness of the model. Finally, in Section 4.6, we compare several methods to demonstrate the effectiveness of the proposed model.

4.1. Factors affecting the classification accuracy of long-tailed data

In this section, we explore the factors that affect the classification accuracy of long-tailed data. We conduct a benchmark experiment using the VOC-LT dataset and BM-ResNet to identify specific factors affecting the accuracy of long-tail data.

Table 2 is sorted by the number of samples of each class, which lists the error rate of each class in the test set. Table 3 lists the error rates of local classes. We can obtain the following conclusions:

(1) The class *Person* has the highest number of samples, and the class *Table* has the lowest number of samples, which the error rate of the class *Person* find to be lower than that of the class *Table*. It shows that the number of samples and the error rate are correlated that a lower number of samples usually corresponds to a high error rate.

(2) The total number of samples in classes 1 to 5 is larger than that in classes 6 to 10, and the error rate is the opposite. In addition, we observe that the error rates of class 11 to 15 and class 16 to 20 are 63.33% and 57.09%, respectively. Both error rates are higher than the former 37.01% and 57.09%. With the decrease in the number of samples, the error rates per five classes increase relatively.

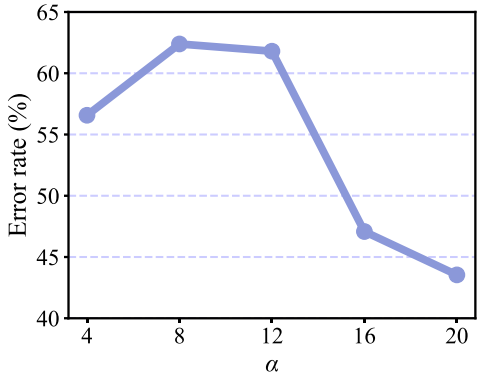

Fig. 6. The error rate corresponding to different number of tail classes.

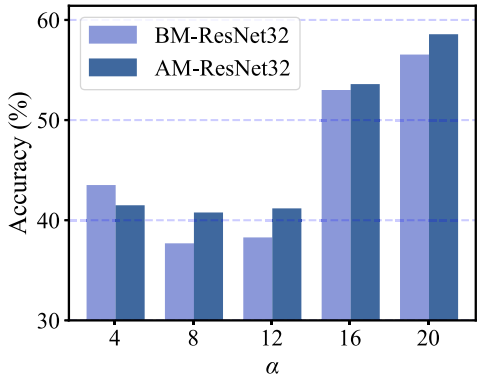

Fig. 7. Comparison of accuracy between BM-ResNet and AM-ResNet in different α (i.e., number of tail classes) on the VOC-LT dataset.

Fig. 6 shows the error rate of different α (i.e., number of tail classes), and we can obtain the following observations:

(1) The error rate of the entire test set is lower than other candidate values when α is 20. Similarly, we find the highest error rate when α is 12. The reason is that there are few samples in the training phase, and the model cannot learn abundant information on sample features.

(2) The error rate decreases when α is from 12 to 16. In addition, the model learns more feature information for samples of different classes with the increased number of samples in the training stage. Therefore, the model improves the classification ability.

According to the above analysis, two main factors affect the classification accuracy of long-tailed data. On the one hand, the lack of training data makes it difficult to train the model sufficiently, making it difficult to achieve the ideal level of feature learning. It affects the generalization performance of the whole model. On the other hand, it is found that when the classical network extracts sample features, some background redundant feature information irrelevant to classification is extracted. Thus, the subject feature information that is easy to classify is weakened, and the classification accuracy is affected.

4.2. Effectiveness comparison of BM-ResNet and AM-ResNet

In this section, we mainly compare BM-ResNet and AM-ResNet, then explore the validity of the AM-ResNet. A comparison of accuracy between BM-ResNet and AM-ResNet in different α on the VOC-LT dataset is shown in Fig. 7. According to the results, we can obtain the following observations.

(1) Parameter α is 20 that represents the global classification accuracy. The classification accuracy of AM-ResNet is about 2.03% higher than that of BM-ResNet. Therefore, AM-ResNet can emphasize effective feature information (i.e., features related to classification) and suppress invalid feature information (i.e., image background feature information).

(2) Parameter α is 8, 12, and 20, representing the local classification accuracy. The accuracy of AM-ResNet is higher than that of BM-ResNet, which is 2.01%, 3.9%, and 0.59%, respectively. The results show that AM-ResNet is effective in improving classification accuracy.

(3) The AM-ResNet accuracy does not improve when samples are very few. The reason is that because there are too few tail samples, the model cannot fit the tail sample data well, resulting in a decline of accuracy. As the number of samples

Table 4
Comparison of each class accuracy between BM-ResNet and AM-ResNet on the VOC-LT dataset (%). (Best results are marked in bold.)

(a) Class 1-10

Net \ Class	Person	Cat	Dog	Bird	Aeroplane	Car	Train	Boat	Horse	Sheep
BM-ResNet	71.79	58.53	47.11	58.94	79.78	**65.74**	**69.23**	**68.12**	26.15	**46.15**
AM-ResNet	**74.91**	**63.51**	**47.89**	**64.81**	**83.15**	60.19	63.08	65.22	**33.08**	44.62

(b) Class 11-20

Net \ Class	Cow	Bicycle	Bottle	TV	Chair	Motorbike	Bus	Plant	Sofa	Table
BM-ResNet	30.89	**56.12**	13.27	55.79	28.24	41.56	**74.65**	24.24	**35.42**	0.00
AM-ResNet	**39.84**	51.02	**19.39**	**60.00**	**36.47**	**46.75**	64.79	**33.33**	29.17	0.00

increases, the accuracy increases most when α is 8 and 12. More importantly, the increase in local accuracy is higher than that of global accuracy.

A comparison of each class accuracy between BM-ResNet and AM-ResNet on the VOC-LT dataset is listed in Table 4. We can observe that AM-ResNet can improve the classification accuracy of most classes. Similarly, the accuracy of the tail class is improved when there are few samples of the tail class. For instance, the classification accuracy of six classes is improved by AM-ResNet in Table 4(b). Therefore, AM-ResNet helps capture the spatial correlation between features, and the generated features can be enhanced. The tail class features enhancement can reduce the dependence of the classifier on head class features.

However, we do not consider adding an attention module to the head data for three main reasons.

(1) From the perspective of the shortcomings of deep learning, the model trained in deep learning performs successfully in balanced distributed data. However, many head class samples do not require much attention, while a small number of tail class samples require attention in long-tailed distribution data. Therefore, such imbalanced distributed data degrade the performance of typical supervised learning algorithms designed for balanced distributed data [2].

(2) From the perspective of sample number, the sample number of the head class is adequate, and the extracted features are enough for effective classification. However, we need to enhance tail class features to achieve relative balance when the number of tail class samples is few.

(3) The problem with the current long-tailed learning is that the tail class has few samples and many classes, which leads to the model being biased toward the head class [44]. Hence, we use the attention module to make the decision boundary fair. It is worth noting that the attention module is not added to the global data or head data. Suppose the attention module is added to feature extraction of global data or head data. However, the feature information of all data is fully mined, and its decision boundary remains unchanged.

To sum up, we propose to add tail classes to AM-ResNet and head classes to BM-ResNet for joint training. Further, we obtain a threshold of 0.6, used for all datasets. Thus, it is crucial to use AM-ResNet as a network branch of HRJBA. The experiment is proved in the fourth section.

4.3. Analysis and selection of the parameter λ

In this section, we discuss the effectiveness of head and tail loss participation and explore the influence of parameter λ. We use the parameter λ to control the influence of the head class learning on the tail class learning. The parameter λ is selected from the candidates {0.9,0.8,0.7,0.6,0.5}. The accuracy of different λ on the VOC-LT dataset is shown in Fig. 8, and we observe the followings.

(1) The accuracy increases from 60.16% to 61.74% when λ reduces from 0.9 to 0.8. By reducing the head loss and increasing the tail loss, the model can be biased towards classifying tail classes in the training phase. It alleviates the problem of model skewing towards the head class, which improves the performance of the tail class.

(2) The accuracy decreases from 61.74% to 59.88% when the parameter λ decreases from 0.8 to 0.5. Since the parameter λ balances the attention of the head and tail classes of the model, the model pays too much attention to the tail class and is not friendly to the head class when the parameter λ is smaller.

Therefore, the fusion of head and tail losses affects classification in the model. The results show that the proportions of head and tail losses in the model reached an appropriate state when λ = 0.8. Similarly, we apply this parameter to all datasets.

4.4. Local classification accuracy

In this section, we discuss the local accuracy results obtained using our method. We compare the classification effects of local classes under different methods. These comparison methods include baseline methods (Baseline), optimal methods (Optimal), and methods that use different attention modules for head data and tail data (HATA). The comparative experiments demonstrate the effectiveness of our proposed method.

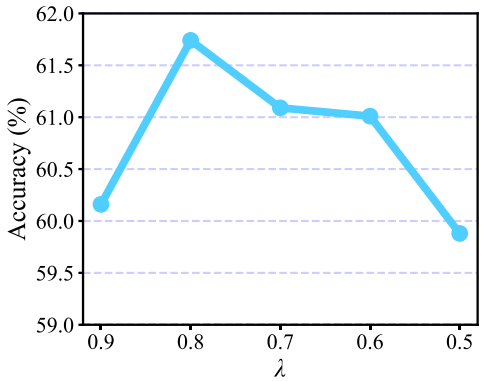

Fig. 8. Accuracy of different λ in fusion of head and tail losses on the VOC-LT dataset.

Table 5
The accuracy (%) of local class on the VOC-LT dataset. (Best results are highlighted in bold.)

Method	Head	Middle	Tail	All
Baseline	63.99	36.50	42.91	56.46
Optimal	65.98	37.26	45.45	58.23
HATA	66.43	**44.27**	52.36	60.58
Our	**68.26**	43.08	**53.82**	**61.74**

Table 6
Different classification methods on the VOC-LT dataset for each class accuracy (%). (Best results are marked in bold.)

(a) Class 1-10

Method	Person	Cat	Dog	Bird	Aeroplane	Car	Train	Boat	Horse	Sheep
Baseline	71.79	58.53	47.11	58.94	79.78	65.74	69.23	**68.12**	26.15	46.15
Optimal	72.71	**64.22**	**56.32**	55.13	80.90	62.04	**71.79**	67.39	38.46	43.85
HATA	74.36	59.24	49.74	**68.62**	**83.52**	70.37	64.62	60.87	**46.92**	65.38
Our	**77.47**	61.61	55.00	68.33	80.52	**72.22**	64.10	64.49	43.85	**75.38**

(b) Class 11-20

Method	Cow	Bicycle	Bottle	TV	Chair	Motorbike	Bus	Plant	Sofa	Table
Baseline	30.89	56.12	13.27	**55.79**	28.24	41.56	74.65	24.24	**35.42**	0.00
Optimal	19.51	**57.14**	19.39	52.63	**31.76**	49.35	71.83	30.30	33.33	0.00
HATA	**36.59**	53.06	**25.51**	51.58	22.35	57.14	84.51	**36.36**	33.33	0.00
Our	32.52	55.10	21.43	43.16	18.82	**59.74**	**90.14**	33.33	29.17	**15.38**

Local accuracy of different classification methods on the VOC dataset are listed in Table 5. The number of head class is 8, the middle class is 7, and the tail classes is 5. We can obtain the following observations.

(1) Compared with the baseline method, our method significantly improves tail class accuracy. The overall and tail class accuracies of the baseline method are 56.46% and 42.91%, respectively, while our method outperforms the baseline method by 5.28% and 10.91%, respectively. The results demonstrate that it is useful to build a hybrid network suitable for tail class classification.

(2) Compared with the HATA method, our method slightly improves the accuracy in the tail class and improves the overall accuracy. Different attention modules are used for head data and tail data, respectively. Although the performance is improved compared with the baseline method and the optimal method, the simultaneous use of the attention module expands both the head class feature space and the tail class feature space, resulting in a relatively unbalanced feature space. Similarly, using different attention modules separately increases the running time.

Each class accuracy of different classification methods on the VOC dataset are listed in Table 6, and we can obtain the following conclusions.

(1) Our method is compared with the HATA method in each class accuracy, and it is found that the accuracy of some classes of the HATA method is higher than our method. The reason is that the HATA method enhances both the head class feature space and the tail class feature space, which cause the method to be still relatively biased towards head class classification.

(2) Our method has an advantage over some poorly classified classes. Table 6(b) shows that the prediction accuracy of class *Table* is 0.00%, while that of our method is as high as 15.38%. The results show that the overall sample accuracy of

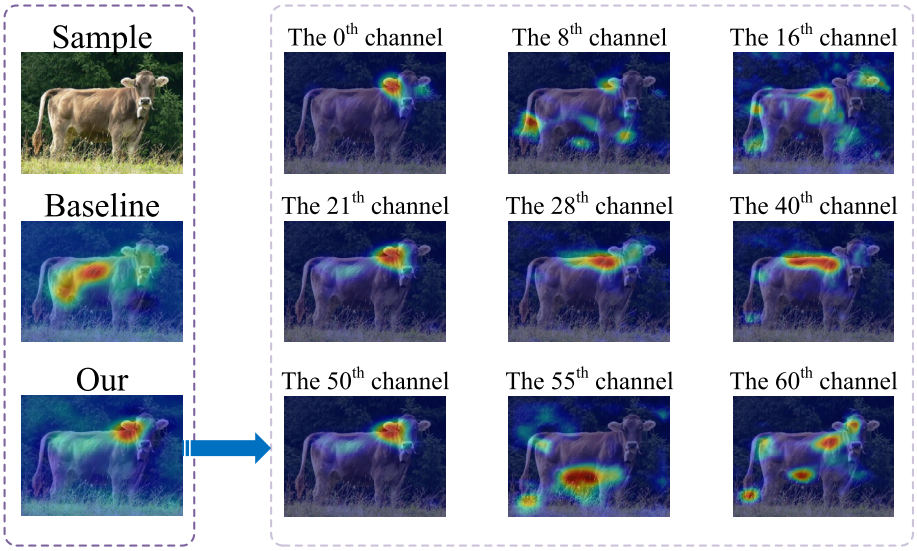

Fig. 9. A visualization of the heat map of the proposed method.

the dataset is higher than that of the other classification methods because the prediction accuracy of most classes is greatly improved.

4.5. Visualization of the proposed model

We use heatmaps as a case to illustrate our motivation and demonstrate model effectiveness. We perform visualization of feature activation by selecting a sample (*Cow*) from the tail class of the VOC-LT dataset. The predicted class scores are mapped to the previous convolutional layer to generate the class activation maps via the class activation map method [45]. Specifically, the activation map is obtained by calculating the weight corresponding to the highest predicted score of the sample and the feature maps of the last convolution layer. Then the activation map is upsampled and mapped to the original image to form a heat map. Similarly, the activation map of each channel is obtained by one-by-one calculation of each feature map of the convolution layer and corresponding weights. The main results are shown in Fig. 9 and we can obtain the following conclusions.

(1) We find that the highlighted areas are dispersed under the trained baseline model. Note that the ventral area of the *Cow* is highlighted. This area is an important image area related to a specific class classification. Nevertheless, this area is ineffective in distinguishing other animals with the same parts. For instance, class *Sheep* and *Cow* are very similar in appearance. The *Sheep* is likely to be misclassified (classified as *Cow*) when entering the classification of the trained benchmark model. Therefore, this area is not the most critical area to differentiate the class *Cow*.

(2) Compared with the baseline heat map, our model can extract more distinguishing and discriminative features. We observe the highlighted areas clustered around the eye and ear under the trained proposed model. The highlighted areas are representative features. Hence, our model can improve the representativeness of the learned tail class features.

(3) To further illustrate the validity of our model, we randomly select the feature map of the last layer and the corresponding weight to calculate and then gain the heat map of each channel. This indicates that our model can remove noise and aggregate non-dispersed effective feature areas.

4.6. Performance comparison with other methods

In this section, we conduct extensive experiments on the VOC-LT, CIFAR-100-LT, and Tiny-imagenet-LT datasets. We further use the datasets to prove the applicability and feasibility of our model. The classification accuracy results for different datasets and methods are shown in Table 7. We can observe from the experimental results that:

(1) Compared with the DRW strategy method, the accuracy of HRJBA using the DRW strategy is improved. In particular, the HRJBA-DRW improves accuracy by 2.86% over LDAM-DRW in the CIFAR-100-LT (ρ =50) dataset. The experimental results show that combining a strategy is effective.

(2) The experimental classification results of our model are generally higher than those of other classification methods. For instance, the accuracy rate of our method is approximately 4% higher than the best classification method on the VOC-LT dataset. It proves that our method can effectively improve the classification ability.

Table 8 shows the experimental results in the Tiny-Imagenet-LT dataset, and we can get the following observations:

(1) Our model achieves 33.81% classification accuracy in the Tiny-Imagenet-LT (ρ =100) dataset. The model is 2.61% outperforms the best model and 2.9% higher than the next best model, which confirms that the model is effective.

Table 7
Comparison of CIFAR-100-LT classification performance under different unbalance ratios (%). (Best results are marked in bold and "-" indicates that no strategy is used during the training phase).

Dataset		VOC-LT	CIFAR-100-LT		
Imbalance ratio		$\rho = 57$	$\rho \approx 50$	$\rho = 100$	$\rho \approx 200$
Loss	Strategy	ACC			
CE	–	56.46	44.57	38.55	34.05
CE	RS	55.86	40.58	33.05	27.76
CE	RW	55.72	39.02	32.94	25.34
CE	CB	54.73	41.44	33.85	23.37
CE	DRS	57.30	47.16	42.16	36.52
CE	DCB	57.67	46.66	41.19	36.72
CE	DRW	57.53	46.79	41.69	36.73
Focal	–	56.77	43.68	39.03	34.68
Focal	RS	54.31	39.66	30.80	27.54
Focal	CB	50.30	37.25	28.70	17.09
Focal	RW	49.59	37.41	29.46	16.39
Focal	DRS	56.68	45.80	41.17	35.75
Focal	DCB	55.50	46.34	40.91	34.58
Focal	DRW	53.77	45.74	40.92	34.85
LDAM	–	58.24	44.91	40.54	37.45
LDAM	RS	54.08	38.07	30.89	25.33
LDAM	CB	51.40	38.75	25.66	22.97
LDAM	RW	52.84	38.88	29.94	21.69
LDAM	DCB	56.68	47.52	43.00	38.69
LDAM	DRS	57.98	47.30	43.53	38.62
LDAM	DRW	58.04	47.59	43.62	38.48
HRJBA	DRW	**61.74**	**50.45**	**44.81**	**40.64**

Table 8
Comparison of Tiny-Imagenet-LT classification performance under different unbalance ratios (%). (Best results are marked in bold and "-" indicates that no strategy is used during the training phase).

Dataset		Tiny-Imagenet-LT		
Imbalance ratio		ρ = 50	$\rho \approx 100$	$\rho \approx 200$
Loss	Strategy	ACC		
CE	–	33.00	28.56	25.60
CE	RS	27.56	24.36	21.37
CE	RW	27.33	21.96	18.46
CE	CB	27.23	21.04	17.73
CE	DRS	34.39	30.11	27.66
CE	DCB	34.38	31.20	26.70
CE	DRW	35.28	30.91	27.69
Focal	–	31.76	28.12	25.56
Focal	RS	27.65	24.17	21.16
Focal	CB	25.13	19.09	12.48
Focal	RW	24.66	18.55	11.15
Focal	DRS	34.11	30.06	26.66
Focal	DCB	33.69	30.53	26.40
Focal	DRW	34.39	30.03	26.35
LDAM	–	30.07	27.61	25.39
LDAM	RS	22.86	19.31	17.14
LDAM	CB	25.99	21.54	15.74
LDAM	RW	25.32	20.68	14.45
LDAM	DCB	33.94	30.37	26.64
LDAM	DRS	32.87	29.79	26.76
LDAM	DRW	33.53	30.54	27.50
HRJBA	DRW	**36.95**	**33.81**	**30.01**

(2) Our model still has good classification ability when there are very few samples of tail class. For instance, the accuracy of our model is 2.31% higher than that of the best model in the Tiny-Imagenet-LT (ρ =200) dataset. It demonstrates that our model can capture classification-related features and enrich feature information of the tail class.

5. Conclusions and future work

In this paper, we proposed a hybrid ResNet based on joint basic ResNet of head classes and attention ResNet of tail classes for long-tailed classification. It can promote the classification ability by enhancing the tail class features. Unlike traditional long-tailed distribution learning, the model fully mined the discriminative feature information of tail classes and did not impair the representative ability of the learned features. In addition, we established a fusion loss function to balance the head loss and tail loss, which improved the classification ability of the model. The experimental results demonstrate that our model is comparable with several state-of-the-art long-tailed distribution learning models. However, we utilized the auxiliary knowledge of long-tailed data (i.e., the samples belong to the head or tail class) for classification in the test stage. In future work, we will focus on the fusion strategy of head and tail features and not use auxiliary knowledge, which solves the long-tailed distribution learning problem.

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.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62141602 and the Natural Science Foundation of Fujian Province under Grant Nos. 2021J011003 and 2021J011006.

References

[1] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[2] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, S.X. Yu, Large-scale long-tailed recognition in an open world, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2537–2546.
[3] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[5] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, L. Van Der Maaten, Exploring the limits of weakly supervised pretraining, in: European Conference on Computer Vision, 2018, pp. 181–196.
[6] 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: IEEE Conference on Computer Vision and Pattern Pecognition, 2018, pp. 8769–8778.
[7] G. Van Horn, P. Perona, The devil is in the tails: fine-grained classification in the wild, arXiv preprint, arXiv:1709.01450.
[8] J. Liu, Y. Sun, C. Han, Z. Dou, W. Li, Deep representation learning on long-tailed data: a learnable embedding augmentation perspective, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2970–2979.
[9] Y. Wan, Y. Han, H. Lu, The methods for moving object detection, Comput. Simul. 10 (10) (2006) 221–226.
[10] C. Papageorgiou, T. Poggio, A trainable system for object detection, Int. J. Comput. Vis. 38 (1) (2000) 15–33.
[11] Y. Ke, L.J. Quackenbush, J. Im, Synergistic use of QuickBird multispectral imagery and LIDAR data for object-based forest species classification, Remote Sens. Environ. 114 (6) (2010) 1141–1154.
[12] Z. Zhang, Y. Xie, F. Xing, M. McGough, L. Yang, MDNet: a semantically and visually interpretable medical image diagnosis network, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6428–6436.
[13] 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.
[14] K.M. Hock, Effect of oversampling in pixel arrays, Opt. Eng. 34 (5) (1995) 1281–1288.
[15] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[16] H. Han, W. Wang, B. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, 2005, pp. 878–887.
[17] S. Barua, M.M. Islam, X. Yao, K. Murase, MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning, IEEE Trans. Knowl. Data Eng. 26 (2) (2012) 405–425.
[18] X. Liang, A. Jiang, T. Li, Y. Xue, G. Wang, LR-SMOTE-An improved unbalanced data set oversampling based on K-means and SVM, Knowl.-Based Syst. 196 (2020) 105845.
[19] S. Yen, Y. Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert Syst. Appl. 36 (3) (2009) 5718–5727.
[20] J. Kim, J. Jeong, J. Shin, M2m: imbalanced classification via major-to-minor translation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13896–13905.
[21] G. Ahn, Y. Park, S. Hur, A membership probability–based undersampling algorithm for imbalanced data, J. Classif. (2020) 1–14.
[22] G. Rekha, V.K. Reddy, A.K. Tyagi, Critical instances removal based under-sampling (cirus): a solution for class imbalance problem, Int. J. Hybrid Intell. Syst. 16 (2) (2020) 55–66.
[23] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156–3164.
[24] L. Yang, R. Zhang, L. Li, X. Xie, Simam: a simple, parameter-free attention module for convolutional neural networks, in: International Conference on Machine Learning, 2021, pp. 11863–11874.
[25] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, X. Wei, Supervised deep feature extraction for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 56 (4) (2017) 1909–1921.
[26] 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.
[27] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[28] H. Zhao, P. Wang, Q. Hu, P. Zhu, Fuzzy rough set based feature selection for large-scale hierarchical classification, IEEE Trans. Fuzzy Syst. 27 (10) (2019) 1891–1903.

[29] H. Zhao, Q. Hu, P. Zhu, Y. Wang, P. Wang, A recursive regularization based feature selection framework for hierarchical classification, IEEE Trans. Knowl. Data Eng. 33 (7) (2021) 2833–2846.
[30] Y. Zhao, W. Chen, X. Tan, K. Huang, J. Xu, C. Wang, J. Zhu, Improving long-tailed classification from instance level, arXiv preprint, arXiv:2104.06094.
[31] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: IEEE International Conference on Computer Vision, 2017, pp. 618–626.
[32] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The Pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[33] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, Technical Report.
[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[35] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, Comput. Sci. 231N 7 (7) (2015) 3.
[36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[37] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, 2016, pp. 630–645.
[38] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, K. He, Accurate, large minibatch sgd: training imagenet in 1 hour, arXiv preprint, arXiv:1706.02677, 2017, 1–12.
[39] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[40] K. Cao, C. Wei, A. Gaidon, N. Arechiga, T. Ma, Learning imbalanced datasets with label-distribution-aware margin loss, arXiv preprint, arXiv:1906.07413.
[41] N. Japkowicz, The class imbalance problem: significance and strategies, in: Artificial Intelligence, vol. 56, 2000, pp. 111–117.
[42] C. Huang, Y. Li, C.C. Loy, X. Tang, Learning deep representation for imbalanced classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5375–5384.
[43] Y. Cui, M. Jia, T. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.
[44] Y. Yang, Z. Xu, Rethinking the value of labels for improving class-imbalanced learning, Adv. Neural Inf. Process. Syst. 33 (2020) 19290–19301.
[45] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.