

## *Retraction*

# **Retracted: Active Learning for Imbalanced Data: The Difficulty and Proportions of Class Matter**

### **Wireless Communications and Mobile Computing**

Received 25 July 2023; Accepted 25 July 2023; Published 26 July 2023

Copyright © 2023 Wireless Communications and Mobile Computing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

### **References**

- [1] X. Cai, "Active Learning for Imbalanced Data: The Difficulty and Proportions of Class Matter," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 4343645, 9 pages, 2022.

## Research Article

# Active Learning for Imbalanced Data: The Difficulty and Proportions of Class Matter

Xinmeng Cai 

School of Computer Science, Wuhan University, Wuhan, 430072 Hubei, China

Correspondence should be addressed to Xinmeng Cai; 2019300003030@whu.edu.cn

Received 23 March 2022; Revised 24 May 2022; Accepted 31 May 2022; Published 29 June 2022

Academic Editor: Haibin Lv

Copyright © 2022 Xinmeng Cai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep learning methods have achieved great success in many fields, but it is severely limited by the quantity and quality of training data. When the number of labeled samples is small and the datasets is class-imbalanced, the model is difficult to perform well. In this paper, an active learning method for the small and class-imbalanced labeled datasets is proposed. In addition to the uncertainty of instances which traditional active learning methods take into account, the difficulty and the proportion of samples of different classes are also considered in our method too, so that our active learning method is better oriented to the dataset with unbalanced classes. This paper first explains the motivation of the proposed method and then introduces the frameworks of the method in detail. Finally, experiments on three datasets prove that our method can obtain better results than traditional active learning methods based on uncertainty.

## 1. Introduction

Deep neural networks have made a great achievement, but they heavily relied on the quantity and quality of training data. On the one hand, annotated data are hard and expensive to obtain so we cannot always train the models with large amounts of labeled instances. On the other hand, in many realistic scenarios, the class distribution of labeled and/or unlabeled data is imbalanced which may cause the models to be biased towards majority classes which have numerous examples and away from minority classes.

Active learning can mitigate the problem of not having enough labeled instances by incrementally selecting samples for annotation that result in high classification performance with low labelling cost. Various solutions have been proposed to help alleviate bias caused by class-imbalanced data, such as resampling [1, 2], reweighting [3, 4], synthesis minority instances [5, 6], and imbalanced semisupervised learning [7, 8]. Above all, these methods are aimed at fully utilizing the information of now labeled and/or unlabeled data, so the best performance of these methods is decided by data

now we have. To increase the upper limit, it is a natural idea to use active learning to find more valuable labeled instances such as querying more instances of minority classes.

The majority of existing AL algorithms rank the value of instances of all classes; this works well when the class is balanced. When facing a situation of imbalanced class, not only the contribution of single instances is different but also that of each class is various. According to the above reasons, in addition to the uncertainty of instances which traditional active learning methods take into account, it is reasonable to also consider the difficulty and proportion of classes. When the difficulty of a class is high, we should query more instances of this class and when the proportion of a class in now training data is small (which means this class is a minority class), we should also query more instances of the class. Through this, we can balance the difficulty and amount of each class and achieve a better performance. The whole procedure can be seen in Figure 1.

We evaluate our methods on the long-tailed datasets which are a common distribution of modern real-world large-scale datasets. The experiment results show that our

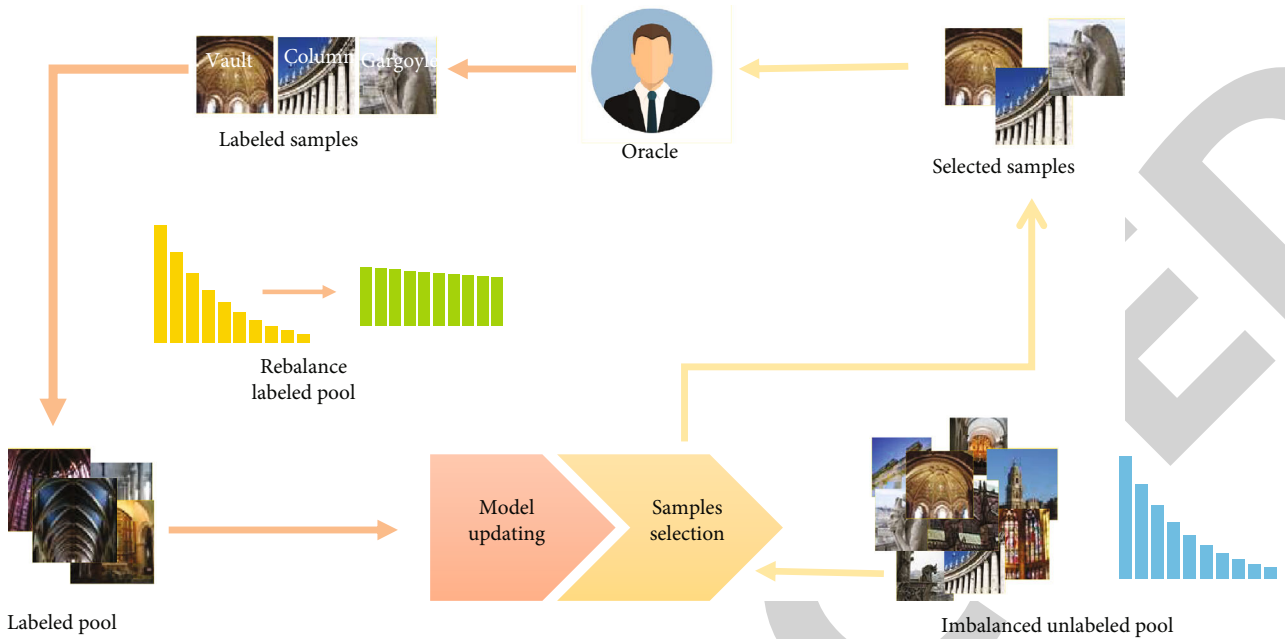


FIGURE 1: Active learning for imbalanced data. At each iteration, active learning rebalances the labeled pool by querying samples for annotation and adding them to the labeled pool.

methods work better than traditional active learning based on uncertainty.

In summary, our main contributions are as follows:

- (i) We proposed a simple method which not only considers the uncertainty of instances but also takes the difficulty and the proportion of classes into account. The only step we add is using the uncertainty score of each instance we already have to compute difficulty of classes. The whole algorithm is simple and easy to complete
- (ii) We introduce inspiration and motivation of our method which explains the reason why we add the difficulty and the proportion of class as evaluation indicators for querying instances and may also offer help for future work
- (iii) We perform experiments on three long-tailed datasets, the results show querying new labeled instances with our methods, and models can achieve higher accuracy than querying with traditional simple active learning which is based on uncertainty

## 2. Related Work

**2.1. Active Learning.** The key idea behind active learning [9] is that a machine learning algorithm can achieve greater accuracy with fewer labeled training instances if it is allowed to choose the data from which it learns. Active learning has been studied for decades, and most of the classical methods can be divided into three categories: (i) membership query synthesis [9–12], (ii) stream-based selective sampling [13, 14], and (iii) pool-based active learning. Nowadays, as the

cost of querying abundant unlabeled samples is cheaper, most of the recent work focuses on the last category.

According to how to evaluate the value of an instance, the pool-based active learning can be grouped into three categories as follows: uncertainty-based methods, representation-based models [15, 16], and their combination [17, 18]. There are three common uncertainty-based methods: least confidence [19], margin sampling [20], and entropy [21]. These three methods measure the uncertainties of novel unlabeled samples from the predictions of previous classifiers and try to find a batch of instances with the highest uncertainties for annotation. Besides these three methods, there are also methods like query-by-committee [22] and error reduction [23]. Our method takes a margin sampling method to evaluate the uncertainty of instances, which is easy to compute. And our method falls into the category of pool-based active learning with uncertainty-based methods.

**2.2. Class-Imbalanced Learning.** There are mainly four categories to solve the problem of learning imbalanced datasets.

**2.2.1. Resampling.** The key idea of this category is to operate the labeled data and make it change to balanced data from imbalanced data. The main methods are oversampling the minority classes [1] and undersampling the frequent classes [2]. The two methods have demonstrated to be helpful for the imbalanced learning. But when the data is limited, we do not have enough samples of frequent classes so it is not wise to discard the frequent classes to balance the class distribution, and the oversampling of minority classes can sometimes lead to overfitting of the minority classes.

**2.2.2. Reweighting.** The normal scheme reweights classes proportionally to the inverse of their frequency [4]. Among

them, some methods [24] focus on rebalancing the contribution of each class and some others [25] focus on reweighting the contribution of each instance.

**2.2.3. Synthesis Minority Instances.** The most typical method is SMOTE [5], and it combines oversampling minority classes and undersampling majority classes. And when oversampling, it involves creating synthetic minority class examples which can achieve better classifier performance. Han et al. improve SMOTE by solving the problem of overlapping among the synthetical samples.

**2.2.4. Imbalanced Semisupervised learning.** Imbalanced SSL use the pseudo-labels to balance class distribution. Yang and Xu [7] argued that leveraging unlabeled data by SSL and self-supervised learning can benefit class-imbalanced learning. Wei et al. proposed CReST [8]. CReST iteratively retrains a baseline SSL model with a labeled set expanded by adding pseudo-labeled samples from an unlabeled set, where pseudo-labeled samples from minority classes are selected more frequently according to an estimated class distribution. In Wei et al.'s paper, they found that the minority classes have high precision which means minority class pseudo-labels are less risky to include in the labeled set. But when the initial sizes of minority classes are too small, even a single mislabeled sample can be devastating to model performance.

**2.3. Class-Imbalanced Active Learning.** Active learning on class-imbalanced data, although a realistic problem, has been under study. Traditional active learning methods are always based on a common assumption: the class distribution of data is balanced. Active learning methods are born to find the labeled instances that the model need, so it is a natural idea to use active learning by querying more samples of minority classes to keep the labeled data balanced. There are few works that focus on the class-imbalanced active learning.

Lin et al. [26] extend the traditional active learning framework by investigating the problem of intelligently switching between asking crowd workers to just label the instances and finding or generating new instances. In other words, the algorithm can order people to search for more needed instances of needed classes using such search engines. But this work may not be so useful when the unlabeled instances are difficult to find, just like the situation when the data are the results of the natural experiment; it is impossible to conduct the experiments again to generate new instances for balance. Zhang et al. [27] proposed BAL-anCe to improve the performance of BALD by employing a novel acquisition function which leverages the structure captured by equivalence hypothesis classes and facilitates differentiation among different equivalence classes. But this method is relatively difficult. Lei et al. [28] improve an active learning method: ATF\* by selecting candidates actively for further annotation from the rank of candidates' information from minidata pool, while keeping the data balance as the original datasets (e.g., 1:1). But this method and its original method focus more on binary classification.

Compared to above all these class-imbalanced active learning, our methods may not defeat them on the aspect of performance, but our method is absolutely easy and can be applied on more situation.

### 3. Main Approach

In this section, we first set up the problem. Next, we introduce the motivation of our methods. Then, we propose our method in detail step by step.

**3.1. Preliminary.** We first set up the problem of class-imbalanced active learning. For an  $L$ -class classification task, there is a labeled set  $X = \{(x_n, y_n) : n \in (1, \dots, N)\}$ , where  $x_n \in R^d$  are training examples and  $y_n \in (1, \dots, N)$  are corresponding class labels. The number of training examples in  $X$  of class  $l$  is denoted as  $N_l$ , i.e.,  $\sum_{l=1}^N N_l = N$ . Besides the labeled set  $X$ , an unlabeled set  $U = \{u_m \in R^d : m \in (1, \dots, M)\}$  and  $M \gg N$ . The initial labeled set is randomly selected from the unlabeled set (typically we assume that the initial labeled set includes at least one instance for all sample, which fits the realistic situation that a classification task would list an instance as an example for all classes).

Unlike the balanced data, our data are class-imbalanced which means the number of unlabeled examples of each class in  $M$  is not the same. Since we randomly select the initial labeled instances, it is of great probability that the number of initial training examples of each class is different too. We measure the degree of class imbalance by the imbalance ratio,  $\gamma = M_{\max}/M_{\min}$ ; the larger the  $\gamma$  is, the more imbalanced the class distribution is.

In our pool-based AL, in each step of the process, a model  $F$  is trained on  $X$  and an acquisition function  $a(U, M)$  chooses  $B$  points to be labeled by an external oracle and added to  $X$ . This process is repeated, training  $F$  with the newly incorporated labeled data, until a certain budget of labeled data is exhausted or until a certain model performance is reached.

**3.2. Motivation.** Previous works [4, 8] introduce long-tailed datasets, and the work [8] mentioned an interesting phenomenon. The model training with class-imbalanced data performs opposite characteristic on minority classes and majority classes, and this is also how the model is biased. They observed that models achieve very high recall on majority classes and poor recall on minority classes, which is consistent with the conventional wisdom.

We conduct experiments on AHE [29] and animals\_10 [30] datasets (we will introduce them in the part of experiments) and found the same result as the work [8] says; for example, the recall of the most majority classes of AHE datasets can achieve 80%, while only about 5% or less samples are successfully recognized by the model. The first and the third plots of Figure 2 can show this.

Despite the low recall, the minority classes maintain relatively high precision. The precision of minority classes may exceed that of majority classes. This is shown in the second and fourth plots of Figure 2. This indicates that many

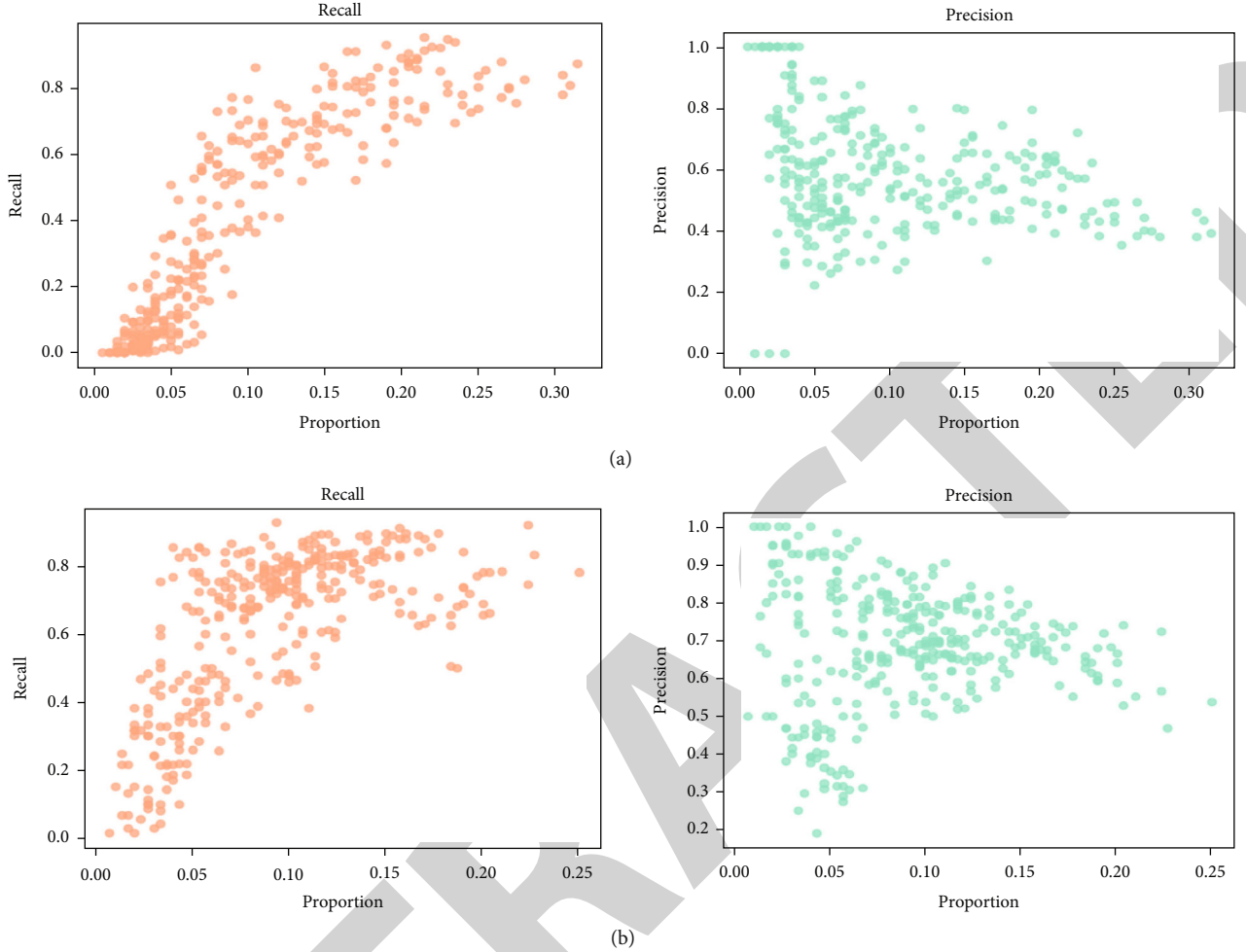


FIGURE 2: Bias of models trained on class-imbalanced data. (a) Per-class recall and precision on animals\_10-LT. (b) Per-class recall and precision on AHE-LT. We train the models for 30 times with different randomly selected training data, and the proportion is the number of each class instance taken in all training data (such that if initially it has 10 instances which belong to the dog class and the whole training data size is 100, then the proportion of the dog class is 10%).

minority class samples are predicted as one of the majority classes.

The relatively high precision shows that if an instance is predicted by the model as a member of minority classes, its ground truth label is very likely to be a minority class. So we can query more instances which are predicted as minority classes and try to make the class distribution more balanced.

**3.3. Instance Uncertainty Estimation.** Our method simply takes margin sampling [20] to estimate the uncertainty of each instance. The selection criteria are based on  $p(y_i = j | u_i; W)$ , which denotes the probability of  $u_i$  belonging to the  $j$ th class.

Margin sampling: rank all the unlabeled samples in a descending order according to the  $ms_i$  value.  $ms_i$  is defined as

$$ms_i = p(y_i = j_1 | u_i; W) - p(y_i = j_2 | u_i; W), \quad (1)$$

where  $j_1$  and  $j_2$  represent the first and second most probable class labels predicted by the classifiers. The smaller the margin is, the more uncertain the classifier is about the sample.

**3.4. Class Difficulty Estimation.** Class-imbalanced data may result in different classes with different difficulty; it is a natural idea to query more instances that belong to a more difficult class. So we estimate the difficulty of classes. After the uncertainty estimation of instances, we get the rank of all the unlabeled samples in a descending order which means the smaller the sorted index is, the easier the instance is. We simply take the most probable class labels predicted by the classifiers as the instance's predicted label. We use  $\hat{u}_i$  to indicate the unlabeled instance in which the predicted label is  $l$  class. And we use  $k_i$  for the ranking of instance  $u_i$ .

Take class  $l$  as an example;  $S_l(i | \hat{u}_i \in S_l)$  denotes the set of instances whose predicted label is  $l$ . We compute the difficulty of class based on the difficulty of the instances

```

1: Input:
   Unlabeled samples  $U$ , initially labeled samples  $X$ , batch size  $B$ ,
   parameters  $\alpha$ , maximum iteration number  $T$ 
2: Output:
   Model parameters  $W$ 
3: Initialize  $W$  with  $X$ 
4: while not reach maximum iteration  $T$  do
5:   Compute  $ms_i$  (the uncertainty of instance) based on (1)
6:   Compute  $Dave_l$  (the difficulty of each class) based on (2), (3)
7:   Compute  $Pro_l$  (the proportion of each class) based on (4)
8:   Compute  $B_l$  (the query number of each class) based on (5), (6)
9:   Query instances from the set  $S_l$  of each class in the descending order according to  $B_l$ 
10:  for each class  $l$  do
11:    query  $B_l$  instances with the lowest  $ms_i$  for labeling
    and add to  $X$  with its annotation
12:  end for
13:  Update  $W$  with  $X$ 
14: end while
15: return Model parameters  $W$ 

```

ALGORITHM 1: Imbalanced active learning.

belonging to this class and use  $Rave_l$  to present the difficulty of class  $l$ .  $Rave_l$  is defined as

$$Rave_l = \frac{\sum_i^{i \in S_l} k_i}{|S_l|}, \quad (2)$$

where  $\sum_i^{i \in S_l} k_i$  is the ranking sum of all the instances whose predicted label is  $l$ , and  $|S_l|$  indicates the number of instances included in set  $S_l$ . The larger the  $Rave_l$  is, the more difficult the class  $l$  is. In other words, the average uncertainty of instances that are included in class  $l$  represents the difficulty of class  $l$ . If the instances of class  $l$  are all of high uncertainty, we can infer that the class  $l$  is of great difficulty.

Furthermore, for normalization of the difficulty, we do more operation. We use  $Dave_l$  to represent the final difficulty of class  $l$ .  $Dave_l$  is defined as

$$Dave_l = \frac{Rave_l}{Rave}, \quad (3)$$

where  $Rave = \sum_{l=1}^L Rave_l / L$  is the average difficulty of all the classes.

**3.5. Class Proportion Calculation.** We use the proportion for the quantitative metrics for minority classes and majority classes. In the training data, the number of training examples in  $X$  of class  $l$  is denoted as  $N_l$  and the proportion of training examples in  $X$  of class  $l$  is denoted as

$$Pro_l = \frac{N_l}{N}. \quad (4)$$

The smaller the  $Pro_l$  of class  $l$  is, the more instances we want to query for annotation and add to  $X$ .

**3.6. Acquisition Function.** For each class  $l$ , the AL method will query  $B_l$  instances from  $S_l$ , where  $B_l$  is computed based

on the difficulty and proportion of the class. The final score of each class  $l$  is denoted by  $Sc_l$ , and  $Sc_l$  is defined as

$$Pro_l = \left( \frac{Dave_l}{Pro_l} \right)^\alpha, \quad (5)$$

where  $\alpha$  is a hyperparameter and is used for reweighting the score of class, and if the imbalance ratio is high, the  $\alpha$  can be bigger.

For each class  $l$ , the query number  $B_l$  is computed as follows:

$$B_l = \text{round} \left( B * \left( \frac{Sc_l}{\sum_{l=1}^L Sc_l} \right) \right), \quad (6)$$

where  $\text{round}(\cdot)$  means rounding the float to int and if the total number of instances that are already queried reaches  $B$ , the AL method will no longer query more instances.

Among all  $L$  classes, the AL method will query in a descending order according to the  $B_l$ , and for each class, the AL method will choose the instances in  $S_l$  with the highest uncertainty (the smallest  $ms_i$ ).

We summarize the whole algorithm in Algorithm 1.

## 4. Experiments

In this section, we first introduce the datasets and implementation setting and then discuss the experimental results.

**4.1. Datasets.** We first evaluate the efficacy of the proposed method on long-tailed AHE, animals\_10, and natural datasets. On these datasets, training images are randomly discarded per class to maintain a predefined imbalance ratio  $\gamma$ . Specifically, the number of instances of the largest class is  $M_{\max}$  and that of the smallest class is  $M_{\min}$ . The class  $l$  has  $M_l = M_{\max} * \gamma^{-(l-1)/(L-1)}$ , where  $l$  indicates that the

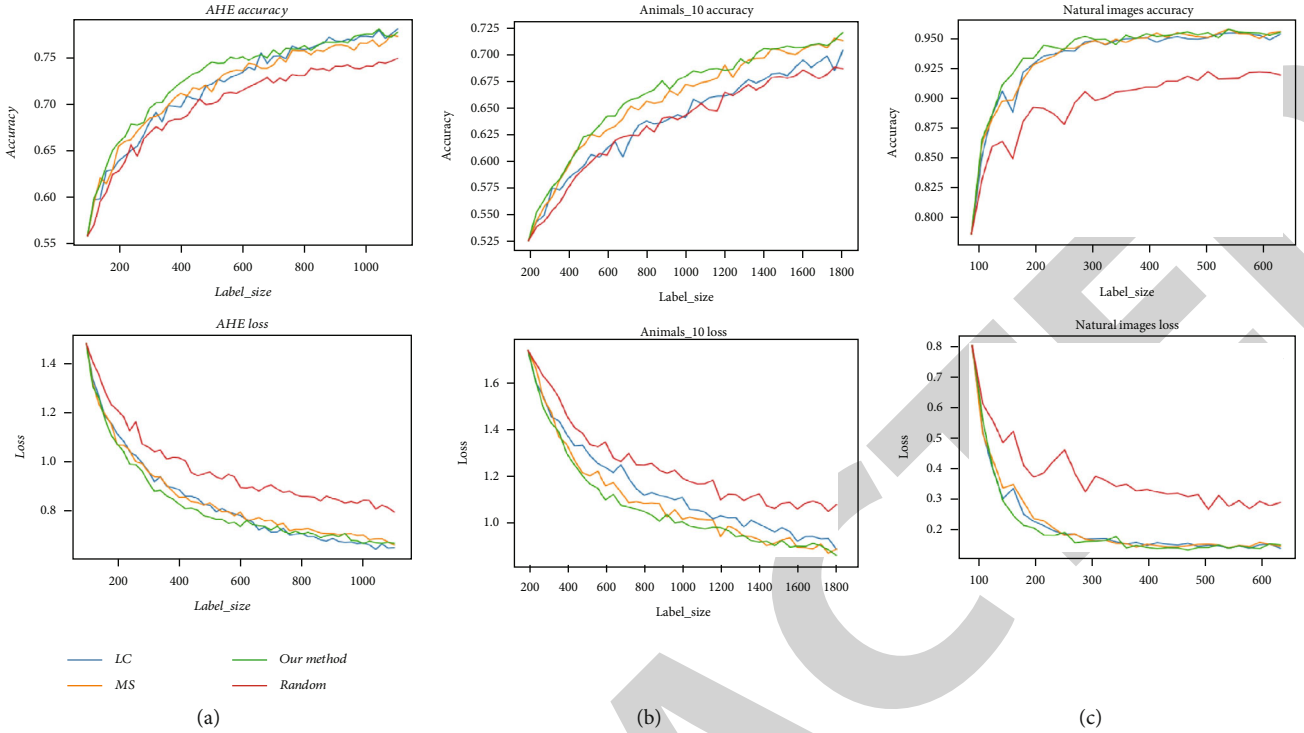


FIGURE 3: Classification accuracy and loss under different datasets: (a) AHE; (b) animals\_10; (c) natural images.

TABLE 1: Accuracy and loss when nine iterations are done on natural images with different methods.

Method	Our method	LC	MS
The highest accuracy	0.93703	0.92230	0.92332
The lowest loss	0.20440	0.23426	0.23964

class is the  $l$ st largest class. And the initial training data is randomly selected from the imbalanced dataset we generated. The test sets remain untouched or randomly select 10% instances per class from the original dataset not overlapping with training data.

**4.1.1. Architectural Heritage Elements Dataset (AHE) [29].** It is an image dataset for developing deep learning algorithms and specific techniques in the classification of architectural heritage images. This dataset consists of 10235 images classified in 10 categories. It is inspired by the CIFAR-10 dataset but with the objective in mind of developing tools that facilitate the tasks of classifying images in the field of cultural heritage documentation. Most of the images have been obtained from Flickr and Wikimedia Commons. We take that the number of instances of the largest class is 1000 and that of the smallest class is 100. The shape of each picture is  $64 * 64 * 3$ . More information can be seen at <https://old.datahub.io/dataset/architectural-heritage-elements-image-dataset>.

**4.1.2. Animals\_10 [30].** It contains about 28K medium-quality animal images belonging to 10 categories: dog, cat,

horse, spider, butterfly, chicken, sheep, cow, squirrel, and elephant. All the images have been collected from “Google Images” and have been checked by human. Image count for each category varies from 2K to 5K units. We take that the number of instances of the largest class is 2000 and that of the smallest class is 200. Since the dataset has not already separated the test set, we randomly select 10% per class for test set. The shape of each picture is  $128 * 128 * 3$ . More information can be seen at <https://www.kaggle.com/alessiocorrado99/animals10>.

**4.1.3. Natural Images [31].** This dataset contains 6899 images from 8 distinct classes compiled from various sources (see Acknowledgements). The classes include airplane, car, cat, dog, flower, fruit, motorbike, and person. We take that the number of instances of the largest class is 900 and that of the smallest class is 100. Since the dataset has not already separated the test set, we randomly select 10% per class for the test set. The shape of each picture is  $100 * 100 * 3$ . More information can be seen at <https://www.kaggle.com/prasunroy/natural-images>.

**4.2. Setup.** We use VGG 16 [32] pretrained on ImageNet as the backbone. We add a fully connected layer with density of 256 and the output layer for the top model. We use Adam [33] as the optimizer. Training continues for 15 epochs for all datasets. And we set the training batch size as 64.

For our method-related hyperparameters, we set  $\alpha = 4/3$  and  $T = 10$ . Note that after 10 iterations, we make a transition between our method and the traditional uncertainty-based method, the part of the batch size is selected by our

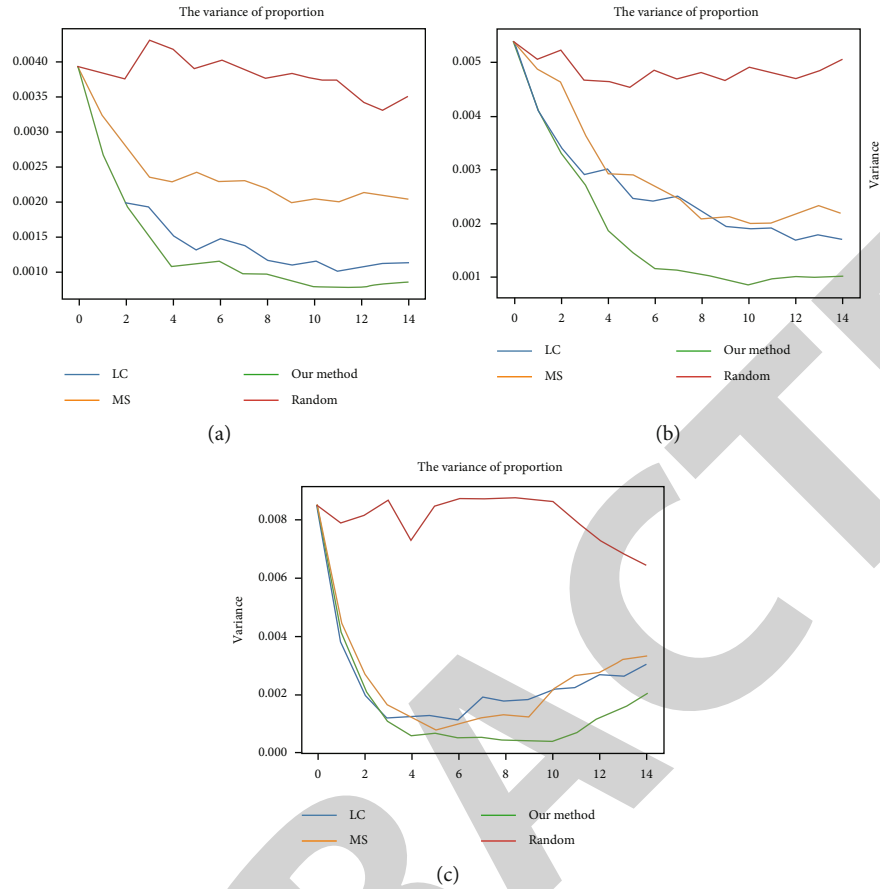


FIGURE 4: The variance of proportion for querying with different methods on three datasets: (a) AHE; (b) animals\_10; (c) natural images.

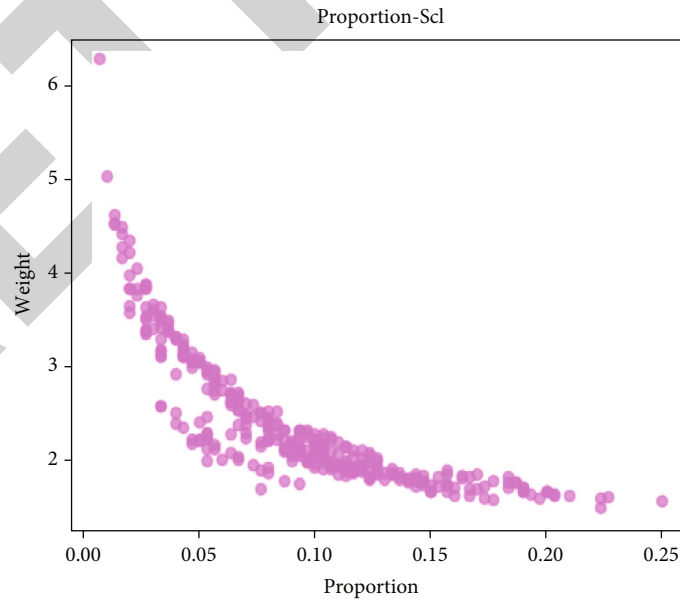


FIGURE 5: The relation between  $Sc_i$  and proportion for each class on AHE.

method, and the other part is selected by the uncertainty method. Because we think that when the performance of a model achieves a relatively high level and when the training

data become relatively big, the influence of imbalanced class distribution will soften. As the iteration grows, the amount of query instances using our method will decrease and at



the same time, the left part of the batch size will query with the uncertainty-based method (margin sampling [20]).

We repeat experiments for 6 times for each dataset, and we report the final performances with the average results of six times.

**4.3. Main Result.** We compare our method with two traditional uncertainty-based methods: (i) least confidence (LC) [19] and (ii) margin sampling (MS) [20], and we also show the results of using random sampling (random). The reason why we do not use entropy [21] is that we found entropy to have lower accuracy than random.

For every dataset, we set the initial training size as 10% of the amount of largest class, and for each iteration, we query 1/5 of the initial label size, which is that if the largest class has 1000 pictures, we set 100 as the initial label size and query 20 pictures each iteration.

As illustrated in Figure 3, we found that our method can achieve higher accuracy than other methods especially at the begging stage. Also, our method can achieve smaller loss than other methods. This result proves that our method can work better on imbalanced-class distribution data.

To show the result more clearly, we also provide the exact accuracy and loss (in Table 1) when nine iterations are done on natural images for six iterations with different methods; we found that our method can achieve about 1% higher accuracy.

**4.4. Extensive Study.** To further study the reason why our method works better on the imbalanced class distribution data, we do some extensive study.

In Figure 4, we show the variance of proportion of class for querying with different methods. The lower variance means that the distribution of class is more balanced. We can see that our method can better rebalance the distribution and this may be an important reason why our method performs better on imbalanced data.

In Figure 5, we show the relation between  $S_C$  and proportion; we find that it shows a relation of negative correlation, which proves that our method for querying more instances of the minority class is reasonable.

## 5. Conclusion and Future Work

In this work, we present an active learning method for class-imbalanced data; our method is motivated by the observation that model training with class-imbalanced data has high precision in minority so that if we query the instances of minority according to the predict result, it is of great probability that we indeed query the instances of minority; thus, we can rebalance the training data. We combine the traditional uncertainty-based active learning method (margin sampling) with our work. We believe that if the class has high difficulty and low proportion, the model will need more instances of this class to improve its performance and lessen the bias of the model. We perform experiments on three datasets and prove that our method can work better on data with imbalanced class distribution.

An important direction for future work is to use a better method for estimating the difficulty of classes. We just simply use the uncertainty of instances that are predicted as the class to estimate the difficulty of classes. In the future, we can combine the representation-based method and uncertainty-based method to better compute the difficulty. Also, we can do more experiments to test when to transfer imbalanced active learning to traditional active learning to get a better performance and how to calculate the hyperparameters instead of setting by people.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The author declares that he has no conflicts of interest.

## Acknowledgments

The work was supported by the Planning Project of Innovation and Entrepreneurship Training of National Undergraduate of Wuhan University: Active online Learning: Method Research and System Application (202110486055). And the author would like to express his gratitude to Mr. Chao Liang who is the tutor of this project.

## References

- [1] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *European Conference on Computer Vision*, Computer Vision – ECCV 2016, pp. 467–482, Springer, Cham, 2016.
- [2] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [3] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, Las Vegas, NV, USA, 2016.
- [4] Y. Cao, T. Chen, Z. Wang, and Y. Shen, "Learning to optimize in swarms," *Advances in Neural Information Processing Systems*, vol. 32, pp. 15044–15054, 2019.
- [5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [6] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, Lecture Notes in Computer Science, pp. 878–887, Springer, Berlin, Heidelberg, 2005.
- [7] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19290–19301, 2020.
- [8] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang, "CRST: a class-rebalancing self-training framework for imbalanced semi-supervised learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10857–10866, 2021.

- [9] B. Settles, "Active learning literature survey," *Computer Sciences Technical Report University of Wisconsin-Madison*, vol. 1648, 2009.
- [10] L. Wang, X. Hu, B. Yuan, and J. Lu, "Active learning via query synthesis and nearest neighbour search," *Neurocomputing*, vol. 147, pp. 426–434, 2015.
- [11] J. J. Zhu and J. Bento, "Generative adversarial active learning," 2017, <http://arxiv.org/abs/1702.07956>.
- [12] Y. Liu, Z. Li, C. Zhou et al., "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1–1528, 2019.
- [13] M. Woodward and C. Finn, "Active one-shot learning," 2017, <http://arxiv.org/abs/1702.06559>.
- [14] C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong, "Stream-based joint exploration-exploitation active learning," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1560–1567, Providence, RI, USA, June 2012.
- [15] R. Urner, S. Wulff, and S. Ben-David, "Plal: cluster-based active learning," in *Proceedings of the 26th Annual Conference on Learning Theory*, pp. 376–397, Princeton University, NJ, USA, 2013.
- [16] O. Sener and S. Savarese, "Active learning for convolutional neural networks: a core-set approach," 2017, <http://arxiv.org/abs/1708.00489>.
- [17] W. Zheng and J. Ye, "Querying discriminative and representative samples for batch mode active learning," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 9, no. 3, pp. 1–23, 2015.
- [18] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, "The power of ensembles for active learning in image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9368–9377, Salt Lake City, Utah, USA, 2018.
- [19] D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," in *17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12, Dublin, Ireland, 1994.
- [20] T. Scheffer, C. Decomain, and S. Wrobel, "The Fourth International Symposium on Intelligent Data Analysis-active hidden Markov models for information extraction," *Lecture Notes in Computer Science*, vol. 2189, pp. 309–318, 2001.
- [21] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.
- [22] R. Gilad-bachrach, A. Navot, and N. Tishby, "Query by committee made real," *Advances in Neural Information Processing Systems*, vol. 18, 2005.
- [23] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 93–102, California, USA, 2019.
- [24] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3573–3587, 2018.
- [25] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, Hawaii · Hawaii Convention Center, 2017.
- [26] C. H. Lin, M. Mausam, and D. S. Weld, "Active learning with unbalanced classes and example-generation queries," in *Sixth AAAI Conference on Human Computation and Crowdsourcing*, Zurich, Switzerland, 2018.
- [27] R. Zhang, A. A. Khan, R. L. Grossman, and Y. Chen, "BAL-anCe: deep Bayesian active learning via equivalence class annealing," 2021, <http://arxiv.org/abs/2112.13737>.
- [28] H. Lei, S. Wang, D. Zheng, X. Qu, S. Fan, and C. Cui, "Improving active learning by data balance to reduce annotation efforts," *The Journal of Engineering*, vol. 2019, no. 23, pp. 8650–8653, 2019.
- [29] <https://old.datahub.io/dataset/architectural-heritage-elements-image-dataset>.
- [30] <https://www.kaggle.com/alessiocorrado99/animals10>.
- [31] P. Roy, S. Ghosh, S. Bhattacharya, and U. Pal, "Effects of degradations on deep neural network architectures," 2018, <http://arxiv.org/abs/1807.10108>.
- [32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <http://arxiv.org/abs/1409.1556>.
- [33] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, <http://arxiv.org/abs/1412.6980>.