CRLT-DD: Condense and Rebalance Long-Tailed Dataset with Dataset Distillation

Keya Hu*
ACM Class
Shanghai Jiao Tong University, SJTU
Shanghai, China
hu_keya@sjtu.edu.cn

Yijin Guo*
ACM Class
Shanghai Jiao Tong University, SJTU
Shanghai, China
guoyijin@sjtu.edu.cn

***Abstract*—Long-tail datasets often result in models overfitting to the head classes (classes with many samples) and performing poorly on the tail classes. In this paper, we aim to apply dataset distillation methods to long-tailed datasets for image classifier problems, which not only improves the training efficiency by reducing the data size but also effectively alleviates the long-tailed problem. With this idea, we propose the strategy called *Condensing and Rebalancing Long-Tailed datasets with Dataset Distillation* (CRLT-DD), using the method called distribution matching for dataset distillation and adding self-supervised pre-train before classifying. We experiment with our approach on CIFAR10-LT and CIFAR100-LT and successfully prove our idea. Our code will be made public.**

***Keywords*-dataset distillation; long-tailed learning; rebalance; self-supervised learning**

I. Introduction

It is universally acknowledged that datasets play a crucial role in Artificial Intelligence. However, there are many problems with datasets in research and experiments. This article pays attention to two specific aspects of them.

For one thing, data naturally follows long-tailed distribution in real-world scenarios, where most of the samples are occupied by a small number of head classes while some tail classes only have a few samples. The imbalanced distribution may cause model predictions to over-bias toward the head classes.

For another, many datasets have large volumes of data, requiring significant time and resources for training. We are looking forward to maintaining sufficient information on the large dataset and achieving the same performance with a smaller dataset. This idea has been introduced and accomplished well with the method of dataset distillation [1].

Therefore, it naturally occurs to us that we can condense some head classes in long-tailed datasets with dataset distillation. Distilling the head classes can help rebalance long-tailed datasets, which is very beneficial for training. However, some methods of dataset distillation themselves hope that the synthetic data can have a similar performance to the original real dataset while some aim for the distilled data to maintain a consistent distribution with the original data. In the context of long-tailed datasets, our final validation indicates that distribution matching is the best approach, aligning with our intuition. Also, the synthetic data from the head classes still maintains more information than the tail classes, which means that our idea still needs experimental verification.

To validate our conjecture, we organize our idea and propose our approach called *Condensing and Rebalancing Long-Tailed datasets with Dataset Distillation* (**CRLT-DD**). Limited by hardware resources, we choose the image classification problem as the subject of our research and only do some lightweight experiments. Through experimental and analytical comparisons among all kinds of existing models, we ultimately utilize the distribution matching [2] as our model of dataset distillation. To reduce the influence of imbalanced distribution, we compress part of the classes in long-tailed datasets to a smaller size and merge the synthetic data with the remaining data into a new dataset. In this method, the distilled size of the compressed classes is determined by the specific characteristics of the datasets.

In this work, we do experiments on CIFAR10 and CIFAR100. We generalize the long-tailed datasets from them respectively with different imbalanced factors and train the classifier network on them. After our distillation, the training results on the synthetic datasets largely outperform the baselines(training directly on the original long-tailed datasets), which reveals that our method can effectively ease the long-tailed problem.

Additionally, we further consider self-supervised learning. We hope that self-supervised pre-train can help mitigate the negative impact of reducing data volume caused by dataset distillation. In our experiment, we choose the SimCLR [3] method for pre-training and preserve the encoder for the final test stage. Our results prove that self-supervised learning is truly positive for training performance.

In summary, our main contributions are as follows.

- For the first time to our knowledge, we utilize the method of dataset distillation to rebalance long-tailed datasets and prove that dataset distillation is an effective way to alleviate the long-tailed problem.
- Our method shows great potential in that it can maintain good training results on long-tailed datasets while effectively reducing the size of the datasets.
- We propose the combination of self-supervised learning

*The first two authors contribute equally.

with dataset distillation, which may become a new pattern for pre-training with further study.

II. Related Work

The majority of data in the world tends to follow a long-tailed distribution and is extensive in size. Constrained by computational resources, it would be very meaningful if the data can be distilled into a smaller set for more efficient training while balancing class data quantity to improve its accuracy. There are recent studies that have investigated training with long-tailed data distributions and explored methods of data distillation.

Deep Long-Tailed Classification. In the natural world, data often exhibits a non-uniform distribution known as a long-tailed distribution. A small portion of the data possesses a large quantity of occurrences, prominent in the distribution and referred to as the "head" or "frequent classes." Meanwhile, the majority of the data comprises significantly fewer occurrences, resulting in a smoother distribution and is referred to as the "tail" or the "rare classes."

This imbalanced data distribution tends to bias classification tasks towards predicting the head data, affecting the accuracy of the larger proportion of tail data. Consequently, various deep learning methods have been applied in classifying long-tailed datasets, including rebalance classifiers, augmentation, self-supervised pretraining and so on. These approaches significantly enhance the accuracy of long-tailed data classification.

Dataset Distillation. Dataset Distillation is a method to use smaller datasets with high information density to reduce resource consumption while preserving model performance. [4] It distills a large amount of source data into a synthesized data set that is significantly smaller than the original data using various methods while attempting to retain as much information as possible from the original data.

To obtain synthetic datasets, there are three mainstream solutions: performance matching, parameter matching and distribution matching. [1], different methods propose different optimization objectives.

Performance matching aims to optimize a synthetic dataset such that neural networks trained on it could have the lowest loss on the original dataset, and thus the performance of models trained by synthetic and real dataset is matched. [1] The method is first proposed in the work by [5], and is followed by [6] to improve its performance.

Parameter matching's key idea is to train the same network using synthetic datasets and original datasets for some steps, respectively, and encourage the consistency of their trained neural parameters. [1] The approach is first proposed by [7], and is extended by a series of following works [8], [9], [10], [11].

The distribution matching approach aims to obtain synthetic data whose distribution can approximate that of real data. [1] Distribution matching has different ways to get features by [2] and [12].

Our goal is to perform data distillation within the context of long-tailed data. We do not intend for the distilled dataset to retain the performance and parameters of the long-tailed data. Hence, we ultimately opt for a Distribution Matching method that ensures as much consistency as possible in the data distribution of each class. Subsequent experiments have also confirmed that the Distribution Matching method outperforms the other two methods when handling long-tail data. At the same time, it also exhibits significantly faster speeds compared to the other two methods since it does not need meta-learning steps.

III. Method

In this section, we propose a method called *Condensing and Rebalancing Long-Tailed datasets with Dataset Distillation* (**CRLT-DD**) to improve the classification accuracy while reducing the size of the dataset.

After comparing different data distillation approaches, we ultimately adopt the distribution matching [2] method that yields the best performance. We also consider using the self-supervised method to better utilize the original long-tailed data information without increasing the volume of data during training.

The distribution matching part is in III-A, the rebalance long-tailed dataset part is in III-B and the self-supervised method part is in III-C. The process of CRLT-DD is shown in Figure 1.

A. Dataset Distillation with Distribution Matching

We employ the distribution matching method proposed in [2] to perform dataset distillation. It uses an empirical estimation of the MMD [13] to estimate the distance between the real and synthetic data distribution. It also applies the differentiable Siamese augmentation $\mathcal{A}(\cdot, \omega)$ to real and synthetic data that implements the same randomly sampled augmentation to the real and synthetic minibatch in training.

Let $\mathcal{T} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{|\mathcal{T}|}, y_{|\mathcal{T}|})\}$ represents the original dataset with $|\mathcal{T}|$ image and label pairs, $\mathcal{S} = \{(\mathbf{s}_1, y_1), ..., (\mathbf{s}_{|\mathcal{S}|}, y_{|\mathcal{S}|})\}$ represents the synthetic dataset with $|\mathcal{S}|$ synthetic image and label pairs, f_θ represents a neural network with parameters θ, and f_θ denotes the model's prediction for data input x. The optimization problem is formalized as:

$$\min_{\mathcal{S}} \mathbb{E}_{\substack{\vartheta \sim P_\vartheta \\ \omega \sim \Omega}} \| \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \psi_\vartheta(\mathcal{A}(x_i, \omega)) - \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} \psi_\vartheta(\mathcal{A}(s_j, \omega)) \|^2 \tag{1}$$

where $\omega \sim \Omega$ is the augmentation parameter such as the rotation degree and P_θ is the distribution of network parameters.

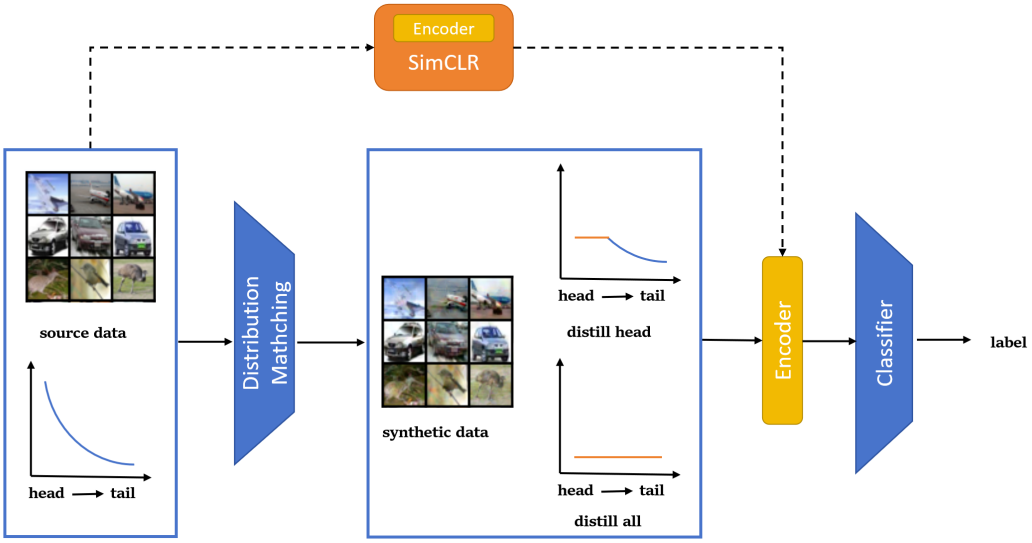


Figure 1. An overview of our CRLT-DD: long-tailed source dataset is condensed by the distribution matching method into synthetic data. One type of condensation is to distill it all into the smallest class size and the other type is to condense only the head of a long-tailed dataset as the figure shows. We also gain an encoder by the self-supervised learning method called SimCLR. Then we use the fixed synthetic dataset and pre-trained encoder to train the classifier.

B. Rebalance Long-Tailed Dataset

We utilize dataset distillation by distribution matching to compress long-tailed distribution data. Long-tailed data typically exhibit a large number of classes at the head, while the data significantly decreases towards the tail, causing classifiers to be biased towards the head and reducing accuracy. Therefore, we attempt data compression to adjust the imbalance between head and tail data.

Let c denotes the number of classes in the dataset, $\mathcal{T} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_c, y_c)\}$ represents the original long-tailed dataset with $|x_1| \leq .... \leq |x_c|$ and $\mathcal{S} = \{(\mathbf{s}_1, y_1), ..., (\mathbf{s}_c, y_c)\}$ represents the synthetic dataset, where $\mathbf{s}_i$ corresponds to the synthesized data for class x_i after dataset distillation.

Distill All Classes Into Same Size. We experiment with compressing all data to the size of the smallest class, ensuring uniform training data across all classes to eliminate the long-tail distribution within the data. The size of distilled data of each class can be represented by the formula:

$$|s_1|, ..., |s_c| = \min_{i \in \{1,...,c\}} |x_i| = |x_1| \tag{2}$$

This method significantly reduces the dataset size while ensuring an equal data quantity for each class.

Distill Head Classes Into Same Size. We observe that some tail classes in the long-tailed data are extremely limited in size, and compressing them to the size of the smallest class might lead to insufficient retention of meaningful information. Instead, compressing only the head data can effectively alleviate the problem of data distribution imbalance. Hence, we also attempt sorting the data based on class size and compressing the larger half of classes to the median size among all classes. We keep the remaining half of the data unchanged, then finally concatenate the distilled head data and unchanged tail data for training. The size of distilled data of each class can be represented by the formula:

$$|s_1|, ..., |s_{\frac{c}{2}}| = |x_{\frac{c}{2}}| \tag{3}$$

$$|s_{\frac{c}{2}+1}| = |x_{\frac{c}{2}+1}|, ..., |s_c| = |x_c| \tag{4}$$

This method achieves a better balance between the imbalanced distributions in long-tailed data without compressing the dataset excessively and losing too much information.

C. Self-supervised Learning for Better Classification

After distilling all classes down to the size of the smallest class, we consider leveraging existing extensive tail data using a self-supervised approach, aiming to obtain well-initialized model parameters through self-supervision to enhance classification accuracy. What's more, by extensively training on a large amount of head data, we aim to learn better feature extraction, consequently improving the classification accuracy of tail-end data. We opt for the self-supervised method utilizing contrastive learning, specifically the SimCLR [3] method by contrastive learning, to perform self-supervision on the tail data.

The encoder from the self-supervision process is preserved after completing self-supervision on the tail data, and subsequent models are initialized with these encoder parameters before training on the distilled data. The encoder will be fine-tuned during the training process of classification tasks.

D. Training Algorithm

We initialize synthetic data with rebalanced size by real image. Train the synthetic data for K iterations. In each iteration, we get the model ψ_ϑ with random parameter $\vartheta \sim P_\vartheta$. Then, we sample a pair of real and synthetic data batch ($B_c^{\mathcal{T}} \sim \mathcal{T}$ and $B_c^{\mathcal{S}} \sim \mathcal{S}$) and map both of them using ψ_θ for every class. Also, sample the augmentation parameter $\omega_c \sim \Omega$ for each class. Calculate the mean discrepancy between the augmented real and synthetic batches of every class after mapping and then sum it as loss $\mathcal{L}$. Update the synthetic data $\mathcal{S}$ by minimizing $\mathcal{L}$. Update the model φ_θ with θ initialized by self-supervised learning for classification using the fixed synthetic data after training.

The experimental results are presented in the following section.

IV. Experiments

We do experiments on two datasets CIFAR10-LT and CIFAR100-LT. We need to rebalance the original long-tailed datasets $\mathcal{T}_{train}$ and generalize synthetic datasets $\mathcal{S}_{train}$. Then we train the classifier network with $\mathcal{S}_{train}$ and test the performance on the datasets $\mathcal{T}_{test}$.

There are three parts in our experiments. Firstly, we attempt different methods of dataset distillation and find out the most appropriate one for our further experiments. Secondly, we record the training results on each class of the datasets and then compare them before and after the distillation, to determine if dataset distillation helps ease the imbalance problem of the long-tailed datasets. Thirdly, we operate our methods on different datasets and take self-supervised methods into consideration.

A. Experiment Setting

Datasets. In all the following experiments, we modify the balanced **CIFAR10**, **CIFAR100** [14] to the uneven setting (named **CIFAR10-LT**, **CIFAR100-LT**). CIFAR10 and CIFAR100 both consist of 60000 32x32 color images. CIFAR10 is labeled with 10 classes with 5000 training and 1000 testing images per class. CIFAR100 has 100 classes with 500 images for training and 100 images for testing per class.

For each dataset, we retain the balanced testing images as $\mathcal{T}_{test}$ and construct long-tailed datasets $\mathcal{T}_{train}$ from the training images by utilizing the exponential decay function:

$$n_i = n\mu^{\frac{i}{C-1}}$$

where C is the number of classes, i is the class index(0-indexed), n is the original number of training images in each class and n_i is the number of images in long-tailed datasets in the i-th class. $\mu = \beta^{-1} \in \{0, 1\}$ and the imbalanced factor β is defined by $\beta = N_{max}/N_{min}$, which reflects the degree of imbalance in the data. The Figure 2 shows the distribution of CIFAR-10-LT with $\beta = 100$. This method is in [15]. In our experiments, we may change β to evaluate our method.

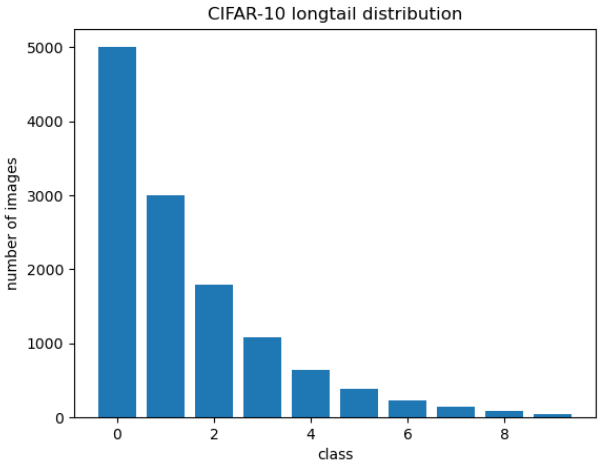


Figure 2. The number of images in each class of CIFAR10-LT with $\beta = 100$

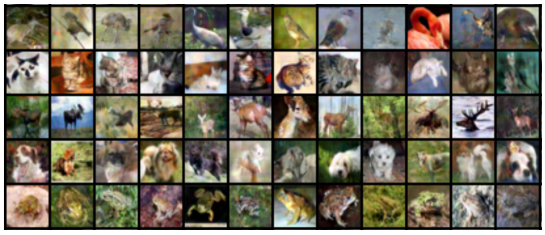


Figure 3. The visualization of synthetic data after dataset distillation

Evaluation protocol. For each dataset, after we distill on $\mathcal{T}_{train}$ and obtain the synthetic data $\mathcal{S}_{train}$, we train a ConvNet on $\mathcal{S}_{train}$ for 500 epoches and test it on $\mathcal{T}_{test}$. The learning rate is 0.01 and the batch size is 128. We report the accuracy to measure the performance.

In all the experiments, we repeat the procedures including distilling, training and testing for 3 times and average the results. We also repeat the testing experiment five times in each procedure and take the average results to avoid large errors.

Some of the learned synthetic images of CIFAR10-LT are visualized in Figure 3 as an example. More details will be mentioned in the following sections.

B. Different Dataset Distillation Methods

We try three methods of distilling, performance matching [5], parameter matching [7] and distribution matching [2] on CIFAR10-LT with $\beta = 100$ in this part. The codes we use are identical to the articles. Performance matching and parameter matching are both meta-learning-based methods. As for the training network in their inner loop, performance matching uses AlexCifarNet [16] while parameter matching utilizes ConvNet, which is consistent with their respective

Table I
DIFFERENT DATASET DISTILLATION METHODS

Method	performance	parameter	distribution
Accuracy(%)	50.35	47.66	**52.37**

Table II
RESULTS ON EACH CLASS

Classes	baseline	distill all	distill head
airplane	0.949	0.719	0.855
automobile	0.943	0.75	0.906
bird	0.718	0.453	0.654
cat	0.636	0.438	0.582
deer	0.569	0.488	0.761
dog	0.355	0.472	0.58
frog	0.446	0.625	0.685
horse	0.337	0.653	0.527
ship	0.135	0.679	0.548
trunk	0.042	0.538	0.201
average	0.513	0.5815	0.6299

original article.

We distill all the classes to 50 images per class (Image Per Class = 50, ipc for short). The testing results are as Table I.

In the results, distribution matching achieves the highest accuracy of 52.37%, which meets our expectations. Moreover, distribution matching does not require meta-learning, resulting in significantly faster training speeds compared to the other two methods.

In fact, the thoughts of distribution matching itself are more aligned with our goal. Performance matching and parameter matching both depend on training. Performance matching targets to get similar performance on synthetic data and real data with the same model, while parameter matching aims to obtain the same parameters in the model during training. This means that their synthetic data still retains the drawbacks of long-tailed datasets.

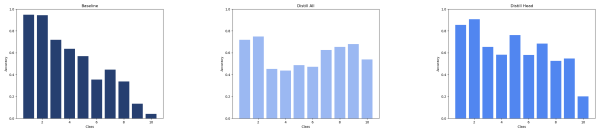
As a result, we choose distribution matching as our distillation method in our following experiments.

C. Results on each class

From now on, we fix distribution matching as our distillation method and update the synthetic data for 4000 iterations. The learning rate is 1.0 and the batch size is 128. Now we utilize CIFAR10-LT with the imbalanced factor $\beta = 100$. In this part, we want to record the results of each class before and after distilling to verify our hypothesis that distillation can help mitigate the imbalanced problem of long-tailed datasets.

The baseline is training directly on the long-tailed dataset $\mathcal{T}_{train}$ without distilling. Additionally, we use two strategies of distillation:

- Distill all the classes to the smallest size. For CIFAR10-LT ($\beta = 100$), this means $ipc = 50$.
- Distill half of the classes to a median size. For CIFAR10-LT ($\beta = 100$), we choose $ipc = 500$.



(a) baseline (b) distill all (c) distill head

Figure 4. Accuracy for each class

We record the test results of each class in Table IV-B and visualize them in Figure 4.

The results of the baseline show us the severe over-bias towards head classes. From the results, we can easily find that dataset distillation can greatly ease the imbalance problem.

It can also be observed that compared to the baseline, distilling all data tends to balance the accuracy across all classes. However, it also slightly decreases the accuracy of the head data. On the other hand, distilling only the head data allows for retaining more data volume within the head, leading to an increase in the accuracy of the head data. Nevertheless, it is noticeable that the accuracy of the tail data decreases in this scenario.

By distilling only the head data, it balances the data volume among different classes while preserving the original dataset's size to a greater extent. As a result, in subsequent testing, it achieves an overall better accuracy.

D. More results and self-supervised

In this part, we operate on CIFAR10-LT and CIFAR100-LT with different imbalanced factors: $\beta = 10, 50, 100$. We utilize three distilling strategies as below:

1 Distill all the classes to the smallest size, named as "distill all".
2 Distill half of the classes to a median size and concatenate it with the remaining unchanged half of the classes, named as "distill head".
3 Firstly use SimCLR for pre-train and preserve the encoder, then distill all the classes to the smallest size named "distill all with SimCLR".

Since our synthetic data is initialized from real images in the distillation, we can record the evaluation results before the 4000 iterations as a baseline for each strategy. This is named a "sample" in Table III.

In Strategy 1, we also add some basic data augmentation. We first distill all the classes and then conduct some simple data augmentation tricks – like crop, flip, rotation, and so on – on the synthetic data. The size of each class after data augmentation is the same as the median size in Strategy 2. This is named "distill all + aug".

Presented in Table III are our experimental results.

From Table III, we can find that sampling from the real data itself is helpful for the classifier accuracy of all kinds of long-tailed datasets compared with vanilla. Additionally,

Table III
RESULTS WITH DIFFERENT DISTILLING STRATEGIES

	Method	CIFAR-10-LT			CIFAR-100-LT		
		IF = 10	50	100	IF = 10	50	100
0	Long-tailed Data Vanilla	71.63	57.29	51.20	39.26	29.91	26.75
1	Sample All	72.45	66.25	49.94	33.40	26.07	12.10
	Distill All	73.15	68.36	58.46	38.89	34.16	19.77
	Distill All + Aug	74.87	63.10	59.54	39.76	23.77	19.34
2	Sample Head	77.89	67.50	62.03	41.58	30.29	28.32
	Distill only Head	**78.26**	68.08	**62.45**	**43.31**	33.65	**30.48**
3	Sample All with SimCLR	72.78	67.31	52.01	35.35	28.82	14.33
	Distill All with SimCLR	73.6	**68.74**	59.32	39.98	**35.12**	18.2

the more imbalanced the dataset is, the greater improvement our method can bring about. This phenomenon proves that dataset distillation is a great method for rebalancing the long-tailed datasets as well as maintaining the initial information from the head classes.

Table IV
THE PERCENTAGE OF THE DISTILLED DATA SIZE COMPARED TO THE ORIGINAL DATA SIZE USING DIFFERENT DISTILLATION METHODS.

origin/distilled %	Method	Baseline	Distill Head	Distill All
CIFAR10-LT	IF=10	100.00	36.52	25.55
	50	100.00	33.65	15.72
	100	100.00	31.77	4.61
CIFAR100-LT	10	100.00	34.00	24.47
	50	100.00	30.59	14.69
	100	100.00	27.33	4.03

Table IV illustrates the proportion of data remaining after distillation using two different methods compared to the original long-tailed dataset size. It shows that distilling only the top data leaves approximately 30% remaining while distilling all data to the minimum class results in less than 25% remaining in CIFAR100-LT and less than 30% in CIFAR10-LT. This significant reduction in data volume while improving classification accuracy largely improve the efficiency of the training. Moreover, in the case of IF=50, CIFAR100-LT even achieved the best accuracy results with only 15% data left, underscoring the effectiveness of this approach in reducing data volume and enhancing accuracy.

It is worth mentioning that CIFAR10-LT and CIFAR100-LT are relatively small-scale datasets. Therefore, the significant reduction in the size of the datasets caused by dataset distillation will to some extent limit the performance of the training results. This is one of the important reasons that distilling only the head classes usually outperforms distilling all whether it uses SimCLR or not.

As for the self-supervised method, it can be observed that the results from random sampling have shown improvement from Table III after implementing self-supervised methods. Moreover, for experiments with distilling all classes by distribution matching, except for the CIFAR-100-LT scenario with IF=100, there's a near one-percentage-point enhancement observed in all other experiments. These findings indicate that employing self-supervised techniques successfully leverages existing data without increasing the training dataset size, leading to improved accuracy. This approach appears to be a promising method to complement data distillation for handling long-tailed data.

V. FUTURE WORK

Several promising directions can be pursued for deeper research and validation in the future.

Firstly, We can attempt a more refined approach to rebalancing the data. Currently, we have only tried distilling all data to the size of the smallest class and distilling the head to a median size among all the classes. However, we can also experiment with altering the number of classes distilled in the head and the data size they are distilled into to achieve better classification results.

Secondly, we can attempt to validate our experiments on a larger dataset, such as ImageNet-LT. When the dataset scales significantly, the superiority of condensing and rebalancing data can be more comprehensively demonstrated. With most of the data retained and different classes balanced, we believe this approach can significantly reduce the data volume and computational resources required while enhancing classification accuracy.

Thirdly, in our experiments, we only tried the SimCLR method as a self-supervised technique. There are many other self-supervised methods worth exploring. At the same time, we believe that data augmentation methods are good ways to balance long-tailed data. By distilling the head data and augmenting the tail data, experimenting with different data augmentation techniques might achieve better classification results.

VI. CONCLUSIONS

In this paper, we present CRLT-DD to condense and rebalance the long-tailed dataset with the dataset distillation method called distribution matching. We try to condense all of the source long-tailed datasets into the same size and condense only the head of the source long-tailed dataset into the same size. We also try to add a self-supervised method SimCLR. We do experiments on CIFAR10-LT and CIFAR100-LT. The accuracy of each class shows that CRLT-DD is a good way to reduce the long-tailed effect and overall

improve the accuracy of classification while largely reducing the size of the dataset, which improves the efficiency of training. The results also show that SimCLR is a good way to improve overall accuracy without increasing the amount of data. We also notice there are some further works we can do in the future, such as altering the size of the condensed dataset, doing experiments on a larger dataset, and trying different self-supervised methods and data augmentation methods.

REFERENCES

[1] Ruonan Yu, Songhua Liu, and Xinchao Wang. A comprehensive survey to dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[2] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[4] Zongxion Geng, Jiahui andg Chen, Yuandou Wang, Herbert Woisetschlaeger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. A survey on dataset distillation: Approaches, applications and future directions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

[5] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[6] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[7] Bo Zhao and Hakan Bilen. Dataset condensation with gradient matching. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[8] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 12352–12364, 2022.

[9] Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z. Pan. Delving into effective gradient matching for dataset condensation. *arXiv preprint arXiv:2208.00311*, 2022.

[10] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10718–10727, 2022.

[11] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[12] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. CAFE: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12196–12205, 2022.

[13] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. In *Technical report, Citeseer*, 2009.

[15] Shu Liu Zhisheng Zhong, Jiequan Cui and Jiaya Jia. Improving calibration for long-tailed recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[16] Ilya Sutskever Alex Krizhevsky and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Guang Li, Bo Zhao, and Tongzhou Wang. Awesome-dataset-distillation. https://github.com/Guang000/Awesome-Dataset-Distillation, 2022.

[19] Yue Xu, Yong-Lu Li, Kaitong Cui, Ziyu Wang, Cewu Lu, Yu-Wing Tai, and Chi-Keung Tang. Distill gold from massive ores: Efficient dataset distillation via critical samples selection. *arXiv preprint arXiv:2305.18381*, 2023.

[20] Ziyu Wang, Yue Xu, Cewu Lu, and Yong-Lu Li. Dancing with images: Video distillation via static-dynamic disentanglement. *arXiv preprint arXiv:2312.00362*, 2023.

[21] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020.

[22] Zhengzhuo Xu, Ruikang Liu, Shuo Yang, Zenghao Chai, and Chun Yuan. Learning imbalanced data with vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[24] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Open long-tailed recognition in a dynamic world. *TPAMI*, 2022.

[25] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6908–6917, 2022.