ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230456

H2T-FAST: Head-to-Tail Feature Augmentation by Style Transfer for Long-Tailed Recognition

Ziyao Meng[a,b], Xue Gu[a,b], Qiang Shen[a], Adriano Tavares[b], Sandro Pinto[b] and Hao Xu[a;*]

[a]College of Computer Science and Technology, Jilin University
[b]School of Engineering, Department of Industrial Electronics, University of Minho
ORCiD ID: Ziyao Meng https://orcid.org/0000-0002-7227-9608, Xue Gu https://orcid.org/0000-0002-2016-3726,
Qiang Shen https://orcid.org/0000-0002-2925-2610, Adriano Tavares https://orcid.org/0000-0001-8316-6927,
Sandro Pinto https://orcid.org/0000-0003-4580-7484, Hao Xu https://orcid.org/0000-0001-8474-0767

Abstract. Deep learning algorithms perform poorly on long-tailed datasets because there is insufficient data in the tail classes to recover its original distribution, resulting in an under-representation of the tail classes in the model. In this work, we propose H2T-FAST, a Head-to-Tail Feature Augmentation method by Style Transfer to improve the performance of the tail. H2T-FAST has the following advantages: (1) It is a fast and universal method that acts on the feature space and so, it can be applied to different backbone networks as well as easily integrated into various imbalanced algorithms with stable performance gains; and (2) it is used only in the training phase and therefore, imposes no additional burden on the deep neural network in the testing phase. In particular, we firstly and randomly select the same number of head samples as the tail ones in each training mini batch. Secondly, the style of the head is transferred to the tail to generate new tail data containing the head style, as a way to increase the number of the tail and get better feature representations. We test our methods on several benchmark vision tasks with state-of-the-art performances.

1 Introduction

Deep learning has made significant advances in the field of computer vision [10, 22, 23]. On the one hand, because of the design of complex convolutional neural networks; on the other hand, artificially designed balanced large-scale datasets are essential to success. However, in reality, most data tends to be unbalanced, even with a long-tailed distribution: most classes have few samples, while only a few common classes have sufficient samples. Most models fail in long-tailed recognition tasks because they will favor the head classes, resulting in an under-representation of the tail classes.

Figure 1 clearly shows the problems with the model on the long-tailed distribution. In Figure 1 (a), we first draw the feature visualization of the training data with T-SNE on an artificially constructed long-tailed CIFAR-10 dataset. Numbers 0 to 9 represent different classes from the heads to the tails in order with the number of points representing the sample size. We can find that the head has a large feature space due to a large amount of head data, but the sample points in the tail occupy a limited feature space due to the small number. Because of the huge variance in feature space of head and tail data, the model will prefer to classify more uncertain data into the head classes. In this work, we hypothesize that the head data contains rich information for us with which we can improve the performance of the tail data. We propose H2T-FAST to use the style information of the head data to merge the tail data for generating new tail data. By doing so, the tail class gets a better and larger feature space. Figure 1 (b) shows the feature visualization plotted after randomly generating 10 tail samples for the tail data using H2T-FAST. It is visible that the tail data has a larger feature space and the decision boundary between different classes is also obvious. In particular, the before and after features of one tail class are circled in the figure.

Numerous current works are mainly divided into two categories: re-weighting and re-sampling. The re-weighting approach is mainly an adjustment of the loss function, generally giving lower weights to the head classes and higher weights to the tail classes, thus counteracting the long-tailed effect in reverse. It is easy to implement and flexible but causes an additional burden for the optimization process. The re-sampling approach focuses on undersampling the head data and oversampling the tail data so that the training samples learned by the model are class-balanced. However, the large amount of data with sufficient variance in the head is not fully learned, while the small amount of data in the tail is often learned repeatedly, resulting in underfitting the head data and overfitting the tail data.

An obvious approach of augmenting the tail classes can solve the problem of little diversity and lack of robustness of the tail classes. Liu et al. [16] transfer the intra-class angular distribution learned from head classes to tail classes. Chu et al. [6] leverage the head class-generic information to recover the distribution of tail classes. Although these methods play a good effect, they do not take full advantage of the style information in the head data classes themselves.

To alleviate the above drawbacks, we draw on the concepts of content and style of images in style transfer. The content generally refers to the shape and form of the image, while the style mainly includes texture and color [9], and the style information extracted in the first few layers of the network usually does not include semantic information. So we use the shallow style information of the head data to change the style of the tail data and keep the tail content unchanged to generate new tail data and ensure that the newly generated tail data is still the original tail class(no matter what color something is, its essence is still the same).

In this paper, we propose H2T-FAST as a new feature augmenta-

* Corresponding Author. Email: xuhao@jlu.edu.cn.

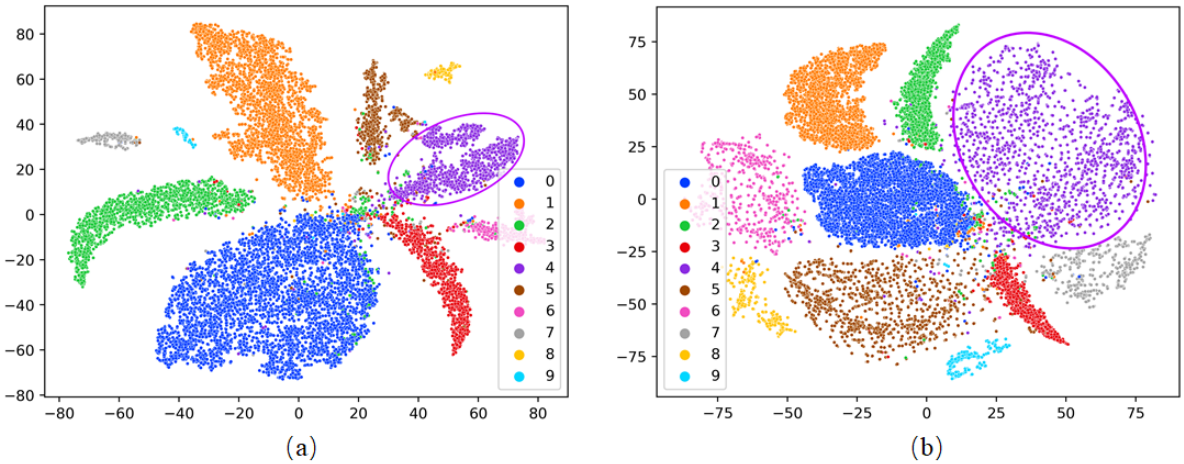

Figure 1. The visualization of H2T-FAST features with T-SNE. (a) The visualization of features from 10 classes in CIFAR10. The tail class has a smaller feature space compared to the head class. (b) We use the H2T-FAST method to get a better representation of the tail features, with wider feature space in the tail classes, especially in the purple class circled in the figure.

tion method. Specifically, we first get the structural style information and the content of the data in some intermediate layer of the network, and afterward randomly fuse the style information of the head data and the content of the tail data to generate new tail data. In doing so, we will get more tail data with different styles and restore the diversity within the class. In summary, our contributions are threefold:

- We propose H2T-FAST, a computationally low-cost feature augmentation based on the style transfer method that increases the amount of the tail and recovers the original tail distribution to improve the performance when training with long-tailed datasets. To the best of our knowledge, it is the first to introduce the style information into the long-tailed recognition and uses the style information from the head to augment the tail classes.
- The proposed H2T-FAST can be applied to all backbone networks and can be integrated into various long-tailed algorithms as well as other data augmentation methods with stable performance gains.
- We evaluate H2T-FAST extensively on various long-tailed settings and confirm that H2T-FAST is universal and effective for different scenarios.

2 Related Work

Re-Balanced Training. For long-tailed recognition tasks, the most widely used solutions are re-weighting [7, 4, 11] and re-sampling methods [26, 12, 3]. Recent studies on re-sampling have yielded good results. BBN [26] dynamically fuses two branches, one learning from the original data and the other learning from the flipped sampled data. Decoupling method [12] uses instance-balanced sampling at the first feature learning stage, and then fine-tunes the classifier with class-balanced sampling.

Various re-weighting methods are designed with complex loss functions. Since the sample size of each class is different, different losses are designed for different classes [17, 25]. In the long-tailed dataset, with the increase of the number of samples, the returns from each sample are significantly diminishing. Therefore, Cui et al. [7] design a better re-weighting method based on the number of valid samples for each class. Jamal et al. [11] combine the above method and adds a conditional weight that requires meta-learning. LDAM [4] encourages larger margins for tail classes and applies re-weighting after normal training. Ren et al. [19] propose a new loss function, the balanced MSE, from a statistical perspective to accommodate long-tailed distribution data.

Re-balanced training method try to rebalance the contribution of each class to the model during the training process. However, these methods will lead to insufficient training of head classes and over-fitting of tail classes to some extent.

Data augmentation. Many data augmentation methods such as Mixup and Cutmix have achieved good results in computer vision. However, using them directly on the long-tailed distribution does not work well. Remix [5] is a simple but effective way to translate the Mixup algorithm to long-tailed distribution by assigning a higher weight to the tail classes. Some methods [16, 13, 5, 18] prove that the information from the head classes can help the tail classes. CMO [18] combines with Cutmix to augment the tail sample using the rich background of the head class as background images. Liu et al. [16] think that the feature distribution was highly correlated with the number of class samples, so the intra-class angular variance of the head class is transferred to the tail to generate new tail data. Chu et al. [6] uses the class activation map to classify the features into class-specific features and class-generic features, and the generated new tail data combines the class-generic features of the head class with the class-specific features of the tail. FASA [24] dynamically generates virtual features to provide more positive samples for tail classes and uses sampling adaptation to avoid over-fitting. MetaSAug [14] augments tail classes with an implicit semantic data augmentation (ISDA) algorithm to ensure that the generated samples contain diverse semantic information.

However, all these methods ignore the style information. Our approach applies style transfer skillfully to the long-tailed distribution by designing the H2T strategy, which can effectively transfer the style information from the head classes to the tail classes. The tail class is augmented by a variety of head style features in order to to make the tail data diverse.

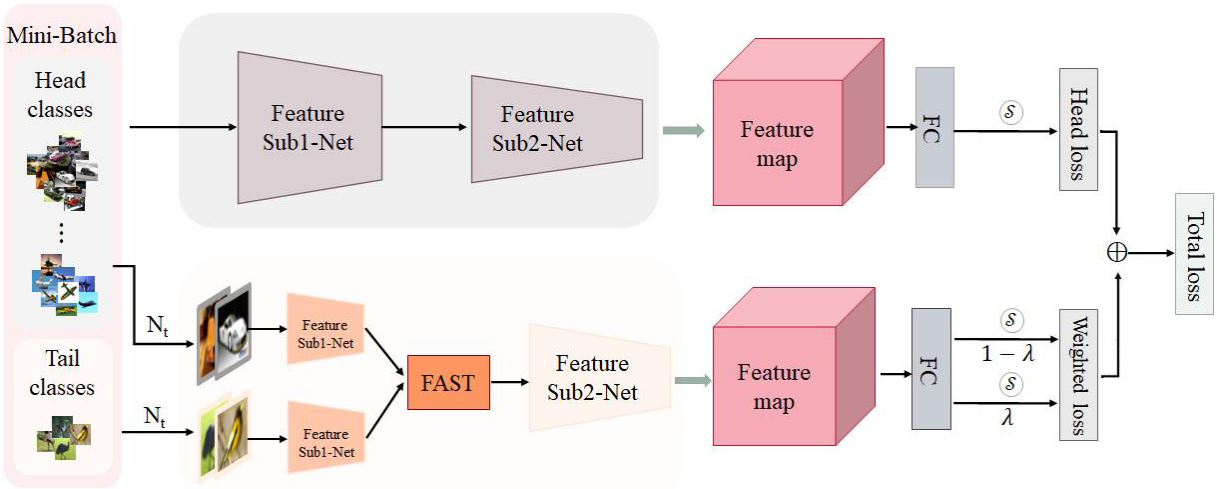

Figure 2. Overview of the H2T-FAST in a mini batch.

3 Method

In this section, we first introduce the overall framework of the H2T-FAST and describe in detail the functionality of each module.

3.1 Overall Framework

Our overall framework shown in Figure 2 consists of two main modules: (1) H2T: Head to Tail strategy, and (2) FAST: Feature Augmentation Based on Style Transfer. Specifically, we first randomly select the same number of head samples as the tail samples in each training mini batch, and after that, the style features of the head and tail samples are exchanged to generate new tail samples that differ significantly from the original tails, and the original distribution of the tails is recovered in this way to obtain better performance.

3.2 H2T

Consider a batch of input samples $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_n}]$, of which we divided into head classes $\mathbf{X_h}$ and tail classes $\mathbf{X_t}$, where the numbers of $\mathbf{X_h}$ and $\mathbf{X_t}$ are $\mathbf{N_h}$ and $\mathbf{N_t}$, respectively. We also take the sum of $\mathbf{N_h}$ and $\mathbf{N_t}$ as equals to the mini batch size. We consider the top 30 percent of the classes with the highest number as the head classes and the rest as tail classes (3 for CIFAR10 and 30 for CIFAR100). The exact threshold of the number of classes will be discussed in the subsequent ablation experiments. After that, we randomly select $\mathbf{N_t}$ samples from the $\mathbf{X_h}$ as $\mathbf{X_h^t}$, the selected samples $\mathbf{X_h^t}$ and tail samples $\mathbf{X_t}$ correspond to each other. In the specific training process, these selected head samples $\mathbf{X_h^t}$ are fed into the model along with the tail samples $\mathbf{X_t}$ to generate new tail samples in the feature space. In short, among a mini batch, when the input is head samples, they will train normally. If the tail samples are inputted to the model, then the H2T strategy is initiated and the head data is randomly selected. The randomly selected head samples will be trained together with the tail samples. We mix the head style feature and tail content feature with the style features of the heads used to extend the tail data. If there is not as much head data as tail data in a batch, then we consider that there is no need to do feature augmentation on the tail data in this batch, the H2T strategy is not needed to be executed and the model will be trained normally.

In addition, to prevent the model from overfitting, we set a hyperparameter p to control the probability of using this module, where p takes values from 0 to 1.

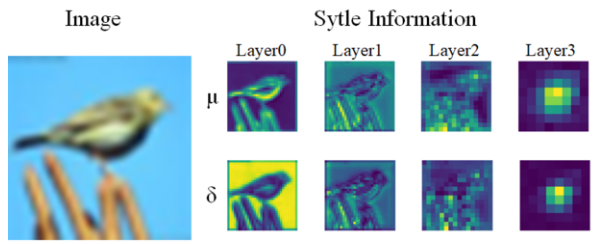

Figure 3. Input image and visualization of feature maps at different network layers.

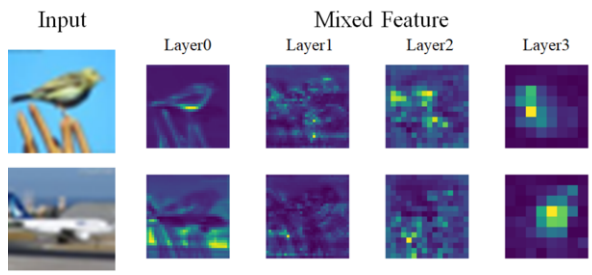

Figure 4. Mix the styles and contents of the two images in different network layers to generate new features. Each line retain the content information of the original image and combine it with the style information of another image.

Algorithm 1 H2T-FAST

Input: Training batch features $\mathbf{X}$.

Output: Head features $\mathbf{X_h}$, Augmented tail features $\mathbf{X_t}$.

```
Splitting batch X into head features X_h and tail features X_t.
for each x_i ∈ X_t do
    Randomly select a sample x_h^t from the X_h
    Extract the style features of x_h^t and the content features of x_i.
    New x_i ← Generate new tail features using style and content
    features extracted from x_h^t and x_i.
    Replace original x_i in X_t with the new tail feature x_i.
end for
return X_h, X_t
```

3.3 FAST

Instead of using the complex encoder-decoder architecture typically needed to extract the semantic style information contained in a style transfer task, we only extract the simple structural and texture style information. The whole process of generating new tail features is shown in Algorithm 1.

As an example, we sample a head sample $\mathbf{x}_h$ and a tail sample $\mathbf{x}_t$ from $\mathbf{X_h^t}$ and $\mathbf{X_t}$ respectively, with $\mathbf{y}_h$, $\mathbf{y}_t$ being their labels. We extract the feature maps of the two samples after an intermediate layer of the neural network, denoted as $\mathbf{f}_h$, $\mathbf{f}_t$. Then we extract two feature maps from a particular dimension to get the corresponding mean μ_h, μ_t and variance σ_h, σ_t. As shown in Figure 3, we show the style information obtained for an image at different network layer. In the first few layers of the network, we can clearly see the style information of the structure and texture of the image from the feature map and becomes less obvious from Layer2. After that, we calculate

$$\mathbf{f}_t^{(h)} = \sigma_h \frac{\mathbf{f}_t - \mu_t}{\sigma_t} + \mu_h. \tag{1}$$

Now, the feature $\mathbf{f}_t^{(h)}$ as a new tail data feature contains not only the content information of the original feature $\mathbf{f}_t$, but also the style information of feature $\mathbf{f}_h$. As shown in Figure 4, we show a series of feature maps of two images mixing their own content and each other's style information at different network layers, and we can clearly see that at Layer0, the aircraft image fuses the bird's style information. The model also has to pay attention to this fused-in information. Therefore, we modify the loss function to predict both the head style and tail content components and design different weights. The tail loss function is

$$L_t = \lambda \cdot \ell(\mathbf{y}_t^{(h)}, y_t) + (1 - \lambda) \cdot \ell(\mathbf{y}_t^{(h)}, y_h), \tag{2}$$

where $\lambda \in [0, 1]$ represents the weights, ℓ represents the set original loss function, $y_t^{(h)}$ represents the output of $f_t^{(h)}$ through the subsequent network and classifier.

The loss in the head branch L_h is the result of a simple loss function calculation. While the final loss L is a simple sum of head loss L_h and tail loss L_t two branches.

$$L = 0.5 \cdot L_t + 0.5 \cdot L_h. \tag{3}$$

4 Experiment

In this subsection, we first introduce the dataset and baseline of the long-tail distribution with the experimental details described and then analyze the experimental results to answer the following scientific questions.

Q1 Whether H2T-FAST gets good results on the long-tailed dataset?

Q2 Whether H2T-FAST is generalized and model-agnostic?

Q3 Whether H2T-FAST enhances other long-tailed algorithms and data augmentation methods?

Q4 Whether H2T-FAST is sensitive to hyperparameters?

4.1 Dataset

We conduct experiments on the artificially created long-tailed CIFAR-10, CIFAR-100, and CINIC-10 datasets with various imbalance factors. In order to compare with other methods, we also tested our method on the step imbalanced dataset [2]. We use standard residual network (ResNet) networks with various depths.

Imbalanced CIFAR. The original version of CIFAR-10 and CIFAR-100 contains 50,000 training images and 10,000 validation images of size 32×32 with 10 and 100 classes, respectively. We reduce the number of training examples per class to create their long-tailed version, keeping the validation set unchanged. We construct two long-tailed versions with ρ of 100 and 10, where ρ represents the ratio of the number which has the highest number of head classes to the number which has the lowest number of tail classes. For the setting of the step unbalanced dataset, we use the setting of μ equal to 0.5 to compare easily with other methods, and μ represents the percentage of tail classes, with the number of head and tail data classes being the same.

Imbalanced CINIC. The CINIC-10 dataset [8] combines data from the CIFAR-10 dataset and ImageNet, with each class containing 9000 images in the training and validation sets, for a total of 10 classes. Since CINIC-10 is constructed from two different sources, it is not a guarantee that the constituent elements are drawn from the same distribution. This property can, however, be leveraged to understand how well models cope with samples drawn from similar but not identical distributions. Furthermore, using the CINIC-10 dataset helps us comparing different methods better because it has 9000 training data, which allows us to conduct extensive experiments with various imbalance ratios while making sure each class still preserves a certain number of data. This helps us to focus more on the imbalance between classes rather than solving a few-shot classification problem for the tail classes. And in this dataset, we explored more imbalanced rates ρ as 200, 100, 50, and 10.

4.2 Implementation details.

For fair comparisons, we use the same setting as used in the past work [4]. In datasets CIFAR-10 and CIFAR-100, we use Resnet-32 as the backbone network, and in dataset CINIC-10, we use ResNet-18 as the backbone network. For both CIFAR-10 and CINIC-10, we train 200 epochs with mini-batch size 128 and decay the learning rate 0.01 at 160, 180 epoch. We use stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay of 0.0002. The difference for CIFAR-100 is that the number of training epochs is 300 and decay the learning rate 0.01 at 150, 225 epochs. We apply the standard data augmentation, which is the combination of random crop, random horizontal flip, and normalization. We use the mean over 5 runs as the final result for all experiment results.

Dataset	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	Long-tailed		Step		Long-tailed		Step	
Imbalance Ratio	100	10	100	10	100	10	100	10
ERM	71.86	86.22	64.17	84.02	40.12	56.77	40.13	54.74
BBN [26]	79.82	**88.32**	-	-	42.56	59.12	-	-
Remix [5]	75.36	88.15	68.98	86.34	41.94	**59.36**	39.96	57.06
LDAM [4]	73.35	86.96	66.58	85.00	39.60	56.91	39.58	56.27
DRW [4]	74.86	86.88	71.60	85.51	40.66	57.32	41.14	57.22
DRS [4]	74.50	86.72	72.03	85.17	40.33	57.26	41.35	56.79
Focal Loss [15]	70.18	86.66	63.91	83.64	38.41	55.78	38.57	53.27
CB Loss [7]	74.11	87.23	65.53	85.7	38.32	55.71	-	-
LDAM-DRW [4]	77.03	88.16	76.92	87.81	42.04	58.71	45.36	**59.46**
H2T-FAST(our)	75.84	87.67	69.46	86.53	41.76	58.17	39.97	56.21
H2T-FAST-DRW(our)	78.36	87.88	75.67	**88.09**	42.66	58.18	43.01	57.62
H2T-FAST-LDAM-DRW(our)	79.88	87.81	78.82	87.94	41.98	57.06	43.74	55.96
HTT-FAST-LDAM-DRW + Mixup(our)	80.95	87.03	78.65	86.99	45.24	57.95	44.9	57.16
HTT-FAST-LDAM-DRW + Cutmix(our)	**81.79**	87.73	**79.77**	87.84	**46.14**	59.32	**46.02**	59.06

Table 1. Top-1 accuracy on long-tailed CIFAR-10 and CIFAR-100.

Imbalance Type	Long-tailed				Step			
Imbalance Ratio	200	100	50	10	200	100	50	10
ERM	56.16	61.82	72.34	77.06	51.64	55.64	68.35	74.16
DRW [4]	59.66	63.14	73.56	77.88	54.41	57.87	68.76	72.85
DRS [4]	57.98	62.16	73.14	77.39	52.67	57.41	69.52	75.89
LDAM-DRW [4]	60.80	65.51	74.94	77.90	54.93	61.17	72.26	76.12
Remix [5]	58.86	63.21	75.07	79.02	54.22	57.57	70.21	76.37
Remix-DRW [5]	62.95	67.76	**75.49**	**79.43**	62.82	67.56	**76.55**	79.36
H2T-FAST-LDAM-DRW(our)	**65.62**	**69.01**	73.16	79.22	**66.04**	**71.1**	73.76	**79.45**

Table 2. Top-1 accuracy on long-tailed CINIC-10 with different imbalance ratio.

Baseline Methods for Comparison. We compare our methods with vanilla training, state-of-the-art techniques, and their combinations.

- Empirical risk minimization (ERM): Standard training method without any strategy.
- Focal Loss: Uses focal loss instead of cross entropy.
- LDAM: Uses label-distribution-aware margin loss instead of cross entropy.
- Re-weighting and deferred re-weighting (RW, DRW): Different weights are assigned to each class of samples. Deferred re-weighting is to use RW after the network has been running for a few epochs.
- Re-sampling (RS): Different sampling probabilities are adopted for different classes.

4.3 Performance Analyses

To verify the question Q1, we did experiments on three datasets. The results of the experiments demonstrate that H2T-FAST achieves the state-of-the-art results.

Long-tailed CIFAR. Table 1 shows that our proposed method H2T-FAST outperforms most of the state-of-the-art methods. In detail, on the CIFAR-10 dataset, the higher the imbalance ratio, the betters the model performance, indicating that H2T-FAST method is better for dealing with extreme long-tail situation. In addition, the H2T-FAST method works better when using Cutmix and Mixup. This demonstrates that H2T-FAST can be well combined with other data augmentation methods, especially the Cutmix method. Moreover, we find the H2T-FAST method is more effective on the CIFAR-10 dataset than the CIFAR-100 dataset, because the number of CIFAR-10 data is fewer, which leads to more head-to-tail data interactions.

Long-tailed CINIC. The results of the CINIC-10 dataset are summarized in Table 2. We found that the higher the imbalance rate or step rate is, the better effect of the H2T-FAST boosting, regardless of whether the data present a long-tailed or step distribution, this finding is also consistent with the results on the CIFAR10 dataset, which further shows the effectiveness of H2T-FAST method. Comparing Table 1 and Table 2, there is an interesting finding: The H2T-FAST method has a more significant effect on the CINIC-10 dataset. This is because the CINIC-10 dataset is a hybrid version of the CIFAR-10 and ImageNet datasets, each class has a large number of samples with high intra-class diversity, the original distribution of the tail classes is better recovered by fusing the style information of the diverse heads. Therefore, when the head and tail samples are mixed, the model will work better.

Normalization	InstanceNorm	BatchNorm	LayerNorm	PONO
Layer0	**75.64**	**75.73**	**75.54**	**75.84**
Layer1	74.05	74.16	73.94	75.81
Layer2	71.75	71.58	75.04	75.81
Layer3	28.85	67.32	67.55	67.11

Table 3. Different Style feature extraction methods in different intermediate layers.

λ	1	0.9	0.8	0.7	0.6
Layer0	**75.84**	73.08	73	71.71	72.33
Layer1	75.81	**75.12**	**74.39**	71.85	72.22
Layer2	75.81	74.87	73.76	**72.71**	**72.96**
Layer3	67.11	69.98	69.44	69.7	70.09

Table 4. PONO in different intermediate layers with various style label weights.

4.4 Ablation Study

With the following ablation experiment, we answered question Q2 and Q3.

Combining different baselines with H2T-FAST. We combine different loss functions and data augmentation methods to verify the effectiveness of H2T-FAST, as shown in Table 5. H2T-FAST has a great improvement in the way of modifying the loss function, especially for focal loss, which has the biggest improvement of 5.18%. And there is also a significant improvement by H2T-FAST to the other data augmentation method. In particular, in combination with the Cutmix method, the use of the H2T-FAST method improves by 2.27% relative to the original method and achieves the best results.

The results of the ablation experiment show that H2T-FAST method is model-agnostic and generic. H2T-FAST method can be combined not only with other long-tailed algorithms but also with data augmentation methods.

CIFAR10	Long-tailed	
	w/o H2T-FAST	w/ H2T-FAST
ERM	71.86	75.84 (+3.98)
DRW	74.86	78.36 (+3.50)
Focal	70.18	75.36 (+5.18)
Focal-DRW	75.60	77.76 (+2.16)
LDAM	73.35	76.99 (+3.64)
LDAM-DRW	77.03	79.88 (+2.85)
Mixup	78.86	80.95 (+2.09)
Cutmix	79.52	81.79 (+2.27)

Table 5. Top-1 accuracy of ResNet-32 with H2T-FAST for different loss functions and augmentation methods on Imbalanced CIFAR10 with $\rho = 100$

4.5 Hyperparameter Analysis

To answer question Q4, we performed several experiments to verify the sensitivity of the model to each hyperparameter.

Choices of extracting style features methods. We explored the effectiveness of different methods for extracting style features at different intermediate layers of the network. Table 3 shows that all methods applied at Layer0 of the network are the best, and PONO [20] is better than the other methods. We hypothesize that the reason is that PONO captures local style information and this style information is a category independent without carrying the original labels, while LN [1], IN [21] and BN compute global features which carry more original information in the later layer of ResNet. In particular, instanceNorm gets the worst results in the third layer of the network, which is due to the fact that the style information carries a lot of labeling information at this point. To verify this idea, we set up another experiment with different weights λ for the style labels. As λ has a lower weight, the fused image contains more labels with style information. As shown in Table 4, with a higher proportion of labels in the style, the performance is better at the later layers of the network. But the overall performance of the network also decreases, which indicates that the deeper the network is extracted the more style information contains the original label information. Therefore, we prove that it is meaningless to fuse the deep label information of the head data to the tail data. So we set λ to 1. The newly generated images still belong to the tail class and have no relationship with the newly added head styles. The shallow style information is not sufficient to represent the whole class.

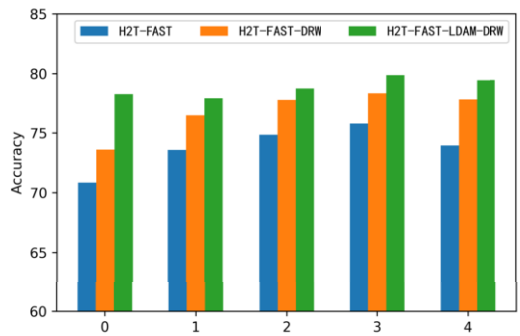

Figure 5. Histogram of the accuracy for the three methods with different head and tail thresholds.

Dataset	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	Long-tailed		Step		Long-tailed		Step	
Imbalance Ratio	100	10	100	10	100	10	100	10
H2T-FAST(our)	0.4	0.2	0.1	0.4	0.3	0.3	0.4	0.4
H2T-FAST-DRW(our)	0.2	0.1	0.1	0.4	0.3	0.3	0.5	0.3
H2T-FAST-LDAM-DRW(our)	0.1	0.2	0.1	0.2	0.5	0.3	0.5	0.5

Table 6. The best probability p on long-tailed CIFAR-10 and CIFAR-100.

Evaluate different thresholds for dividing heads and tails. We test the results on the CIFAR-10 dataset when the head and tail classes are divided by different thresholds. As seen in Figure 5, all hyperparameters obtained better results, and the best one is obtained when the threshold value is 3. So we use thirty percent of the class number as the threshold.

For all step datasets, the number of classes at the head and tail should theoretically be the same. However, in our method, to be consistent with exp datasets, the threshold is set to 3, 30, corresponding to CIFAR-10 and CIFAR-100, respectively.

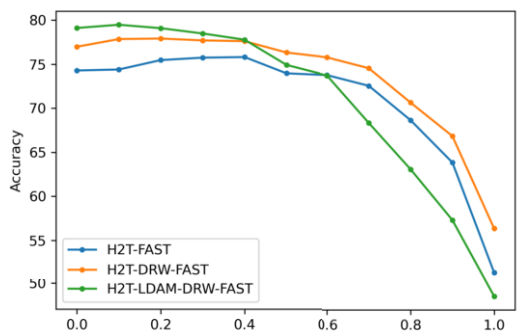

Figure 6. Accuracy of the three methods for different probability p.

Model	p	λ	Top1
ResNet32	0.1	1.0	79.88
		0.9	79.49
		0.8	78.88
		0.7	78.12
		0.6	75.32
		0.5	74.10
		0.4	68.72
		0.3	57.69
		0.2	49.03
		0.1	40.03

Table 7. Accuracy of H2T-LDAM-DRW-FAST on CIFAR-10 with differnet λ

Evaluate different probability p and λ. The Figure 6 and Table 7 show the effect of different p and λ on the experiments, respectively. We find that the performance is good when λ is fixed and p is less than 0.6. This is because with increasing p values, the tail data has a higher probability to exchange, which will lead to over-fitting of the tail data and thus affect the performance of the head classes. This leads to a decline in the overall performance.

We also searched for probability values from 0.1 to 0.5 in all experiments as shown in Table 6. On the CIFAR10 dataset, the best probability is 0.1 or 0.2, while on the CIFAR100 dataset, the best probability is 0.3 or 0.5. We found that the best probability p is relatively large on the CIFAR100 dataset, which indicates that the H2T-FAST should set a higher probability on more complex datasets to allow more interaction between the head class information and the tail class information.

Moreover, we further verify that the larger λ is better on H2T-FAST-LDAM-DRW. When the probability p is fixed, as λ decreases, and the proportion of style tags increases, the overall performance of the network begins to degrade.

5 Conclusions and Future Work

We propose a feature augmentation method for long-tailed recognition, which only performs a small increase in computation during the training process. New tail data is only generated during the training process, where the tail data is combined with the style features of the head in one mini batch to generate new tail data. Our method is simple and effective, and we validate it in various benchmark vision tasks. Furthermore, we have demonstrated the effectiveness of the method by conducting numerous ablation experiments. However, our method only considers the fusion of one image in the head and one in the tail, and methods to fuse more images can be considered in the future. Moreover, because of the need to classify head and tail classes, it is not yet possible to migrate to other image classification tasks with many classes. It would be a good direction for research in the future to design a better strategy for not having to distinguish between head and tail data.

6 Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC), "From Learning Outcome to Proactive Learning: Towards a Human centered AI Based Approach to Intervention on Learning Motivation" (No. 62077027), and the European Union's Horizon 2020 FET Proactive project "WeNet-The Internet of us" (No. 823783) and the work is also supported by the Department of Science and Technology of Jilin Province, China (20230201086GX).

References

[1] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, 'Layer normalization', *CoRR*, **abs/1607.06450**, (2016).
[2] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski, 'A systematic study of the class imbalance problem in convolutional neural networks', *Neural Networks*, **106**, 249–259, (2018).

[3] Jonathon Byrd and Zachary Chase Lipton, 'What is the effect of importance weighting in deep learning?', in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, eds., Kamalika Chaudhuri and Ruslan Salakhutdinov, volume 97 of *Proceedings of Machine Learning Research*, pp. 872–881, (2019).

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma, 'Learning imbalanced datasets with label-distribution-aware margin loss', in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, eds., Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, pp. 1565–1576, (2019).

[5] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan, 'Remix: Rebalanced mixup', in *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, eds., Adrien Bartoli and Andrea Fusiello, volume 12540 of *Lecture Notes in Computer Science*, pp. 95–110, (2020).

[6] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling, 'Feature space augmentation for long-tailed data', in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, eds., Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, volume 12374 of *Lecture Notes in Computer Science*, pp. 694–710, (2020).

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie, 'Class-balanced loss based on effective number of samples', in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 9268–9277, (2019).

[8] Luke Nicholas Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey, 'CINIC-10 is not imagenet or CIFAR-10', *CoRR*, **abs/1810.03505**, (2018).

[9] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge, 'Image style transfer using convolutional neural networks', in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2414–2423. IEEE Computer Society, (2016).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, (2016).

[11] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong, 'Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective', in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 7607–7616, (2020).

[12] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis, 'Decoupling representation and classifier for long-tailed recognition', in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, (2020).

[13] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin, 'M2m: Imbalanced classification via major-to-minor translation', in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 13893–13902, (2020).

[14] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng, 'Metasaug: Meta semantic augmentation for long-tailed visual recognition', in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 5212–5221. Computer Vision Foundation / IEEE, (2021).

[15] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár, 'Focal loss for dense object detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, **42**(2), 318–327, (2020).

[16] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li, 'Deep representation learning on long-tailed data: A learnable embedding augmentation perspective', in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 2967–2976, (2020).

[17] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar, 'Long-tail learning via logit adjustment', in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, (2021).

[18] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi, 'The majority can help the minority: Context-rich minority oversampling for long-tailed classification', in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 6877–6886. IEEE, (2022).

[19] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu, 'Balanced MSE for imbalanced visual regression', in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 7916–7925. IEEE, (2022).

[20] Ryu Takeda, Kazuhiro Nakadai, and Kazunori Komatani, 'Spatial normalization to reduce positional complexity in direction-aided supervised binaural sound source separation', in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2021, Tokyo, Japan, December 14-17, 2021*, pp. 248–253. IEEE, (2021).

[21] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky, 'Instance normalization: The missing ingredient for fast stylization', *CoRR*, **abs/1607.08022**, (2016).

[22] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, 'Aggregated residual transformations for deep neural networks', in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5987–5995, (2017).

[23] Lu Yang, He Jiang, Qing Song, and Jun Guo, 'A survey on long-tailed visual recognition', *Int. J. Comput. Vis.*, **130**(7), 1837–1872, (2022).

[24] Yuhang Zang, Chen Huang, and Chen Change Loy, 'FASA: feature augmentation and sampling adaptation for long-tailed instance segmentation', in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 3437–3446. IEEE, (2021).

[25] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun, 'Distribution alignment: A unified framework for long-tail visual recognition', in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 2361–2370, (2021).

[26] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen, 'BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition', in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9716–9725, (2020).