Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5323327, 8 pages
https://doi.org/10.1155/2022/5323327

Research Article

FreMix: Frequency-Based Mixup for Data Augmentation

Yang Xiu,[1] **Xinyi Zheng**,[2] **Linlin Sun**,[3] **and Zhuohao Fang**[4]

[1]*School of Cyberspace Security (School of Cryptology), Hainan University, Hainan, China*
[2]*Department of Information Science, Beijing University of Technology, Beijing, China*
[3]*School of Economics, Guangzhou City University of Technology, Guangdong, China*
[4]*Faculty of Information Technology, Macau University of Science and Technology, Macau, China*

Correspondence should be addressed to Yang Xiu; 20190581310067@hainanu.edu.cn

Received 2 March 2022; Revised 14 March 2022; Accepted 17 March 2022; Published 14 April 2022

Academic Editor: Kalidoss Rajakani

Deep learning models have attracted tremendous attention in computer vision in recent years, while most of them heavily rely on massive data for training. As one of the solutions to the sparse data problem, data augmentation techniques, such as image translation and rotation, can substantially increase the model's generalization ability and performance. However, on one hand, these approaches primarily work under the pixel domain, which is limited to fully mining and fusing picture data from the frequency viewpoint. On the other hand, the fusion weighting factors are primarily modified in a manual fashion, which increases the application costs in practice. To this end, we propose a novel method termed as frequency-based Mixup (FreMix) that allows images to be fused in the frequency domain and to improve the efficiency of data augmentation by adaptively adjusting the weighting coefficients in this paper. In FreMix, first, a fast Fourier transformation (FFT) is performed on the input image, such that the frequency information rather than raw pixel information can be extracted for further augmentation. Besides, an exploration-exploitation training paradigm is exploited, such that the FreMix can be trained periodically to facilitate learning and avoid manually hyperparameter settings. We conduct comparing experiments on three benchmark datasets including CIFAR, ImageNet, and ILSVRC2015, and the experimental results validate the effectiveness of the proposed method.

1. Introduction

Existing deep neural networks rely on a considerable quantity of data with low confidence due to the enormous series of tests and the risk of overfitting [1]. Data augmentation [2] approaches have been proposed since deep learning [3] models require big data. In the field of data augmentation, horizontal/vertical flip [4], rotation [5], scaling, cropping, clipping [6], panning, contrast, color dithering [7], noise, pixel domain fusion, and other techniques are available. However, scaling distortion distorts the image [8], and most of these methods suffer from the low-accuracy problem, which further limits their applications in practice.

Existing advances show that automatic enhancement and bit-image blending are the more advanced data augmentation techniques. For example, hybrid-based techniques such as Mixup [9] regulate the neural network enhancing the linearity between training samples. The robustness of the adversarial samples is increased and can stabilize the training process for generating adversarial networks [10]. Mixup uses a convex combination of two pairs of instances and labels to train the network to function linearly in the training instances. This simple learning process gives rise to robustness to adversarial examples and improved calibration capability. Mixup can implicitly control the complexity of the model. However, since the coefficients are manually adjusted, there is a problem that the operation is very inconvenient and there is a cost for experimenting with the parameters. Pixel-based blending affects the nature of the pixel-based image, which is not consistent with the pixel problem [11]. And there is also the obvious problem that using Mixup introduces some very unnatural pseudopixel information [12]. To reduce the effect of pseudopixel information, Cutout [13] or CutMix [14] methods are developed, which differ in the pixel values of the filled areas. Cutout enables the model to focus on the regions

(abdomen) that are harder to distinguish from the target, but there is a part of the region without any information, which will affect the training efficiency. Cutout randomly masks the square part of the input image during the training process so the classification algorithm cannot adapt well to the masked data. CutMix uses a region-based enhancement strategy that exploits a binary mask to select the blended regions. Such an operation allows the model to identify two targets from a local view of an image. Also, CutMix allows the learned model to retain a good understanding of the actual data, which can improve the training efficiency and does not have unnatural image blending. However, when CutMix was used to train MobileNetV2 [15] on Tiny imagenet [16], we observed severe performance degradation. Only a tiny section of an image can affect the results in fine-grained image classification tasks. Hence, CutMix blurs the features that are critical for establishing the picture class. This makes the model confusing for classification.

To improve the existing sample blending-based strategy, we propose a frequency-based Mixup (FreMix) which fuses with extracted frequencies and adaptive exploration of the fusion coefficients. Previous related methods are mainly based on the pixel domain mining of the images while lacking the frequency information exploration, which thus leads to incomplete exploitation of the raw data. Thus, in the proposed FreMix, we study a frequency-based method which does not directly sum images in the pixel space but adapts to the interaction of different textures and high and low information of images by fusion in the natural domain of frequencies. Besides, unlike the previous way that requires manual adjustment of parameters, we design an online update method of parameters based on exploration-exploitation mechanism from reinforcement learning, which can solve the problem of manual adjustment of parameters.

Our research was aimed at better mining image data in the field of data augmentation and providing improved methods to avoid manual adjustment of weighting coefficients and also improving the efficiency and accuracy of mining image data by experimentally exploring new solutions to adjust coefficients online. The approach is based on the fusion of natural domains of frequencies to fit the interaction of different textures and high and low information of images, instead of summing directly in pixel space. The automatic hyperparameter optimization is performed using exploration-exploitation mechanism from reinforcement learning. The results of our method on different datasets and different networks improve the accuracy extensively and effectively, such as ResNet [17], VGG [18], and WRN [19]. Our contribution can be summarized as follows: (1) we developed a novel FreMix method in which the frequency domain information of images is leveraged for better data augmentation. (2) An exploration-exploitation mechanism is leveraged in FreMix, such that the proposed method can avoid manually setting hyperparameters and be more applicable in practice. (3) Experiments on three datasets show that the proposed method achieves superior performance in terms of data augmentation.

The remainder of this paper is organized as follows. In Section 2, we present related work in the field of data augmentation and hyperparameter tuning methods. In Section 3, our proposed FreMix is presented with a detailed algorithm procedure provided. In Section 4, the proposed FreMix is verified through two real-world datasets. Finally, Section 5 concludes the paper.

2. Related Work

2.1. Data Augmentation. Data augmentation is a typical machine learning approach that is mainly used to increase the size of the training dataset to make it as diverse as possible so that the trained model can generalize better. Currently, data augmentation mainly includes horizontal/vertical flipping, rotation, scaling, cropping, clipping, panning, contrast, color dithering, and noise. The main existing methods are Mixup, CutMix, and Cutout. Mixup mixes two random samples proportionally, and the proportional distribution of classification results may cause underfitting. Cutout enables the model to focus on the region (abdomen) where the target is difficult to distinguish. Still, there is a part of the region without any information, which will affect the training efficiency. Mixup utilizes all of the pixel data while also introducing some extremely strange pseudopixel data. Mixup uses a convex combination of two pairs of instances and labels to train the network so that the network functions linearly across the training instances. Notably, this simple learning procedure leads to robustness to adversarial examples and improved calibration. AdaMixup [20] diagnoses flow intrusions in Mixup, where a mixed model collides with another example in the data flow, which can cause underfitting. This risk is regulated, and the loss term penalizes the intrusion by an intrusion discriminator. Manifold Mixup [21] uses two intermediate representations as examples at the kth layer. When $k = 0$ implies the input layer, it reduces to vanilla Mixup. CutMix uses a region-based augmentation method that selects the mixing zone using a binary mask for better performance in spatial situations. Note that our approach is related to these methods, but clearly, since we do not use real labels for Mixup supervision. Besides, our proposed FreMix in this paper does not directly sum in pixel space but fits the different textures of images and the interaction of high and low information through the fusion of natural domains of frequencies. It achieves very good performance on several datasets and different models such as CIFAR [22] and ImageNet [23] and greatly exceeds the previous methods.

2.2. Hyperparameter Optimization Method. The setting of hyperparameters has a direct impact on model performance and its importance cannot be overstated. To maximize model performance, it is critical to understand how to optimize hyperparameters. Several standard hyperparameter optimization methods are described. Most of the time, engineers depend on trial-and-error methods which could tune hyperparameters for optimization. On another side, this method is time-consuming and needs a lot of experience. Therefore, many automated hyperparameter optimization methods have been developed. Grid search [24] is arguably the most basic hyperparameter optimization method. Using

this technique, we simply construct separate models for all hyperparameters possible, evaluate the performance of each model, and select the model and hyperparameters that yield the best results. This method enables training for individual hyperparameter combination models and evaluating the performance of each. Each model is independent, so it is easy to perform parallel computation. However, the fact that each model is independent also results in no guidance between models, and the computational results of the former model do not influence the choice of hyperparameters for the latter model.

In contrast, Bayesian optimization methods [25] can draw on existing results to influence the selection of hyperparameters for subsequent models. Only combinations of hyperparameters that are likely to increase model performance are computed, which reduces the number of calculations required for model training assessment. Bayesian optimization works by establishing a posterior distribution of the function that best describes the function to be optimized. The posterior distribution improves as the number of observations grows, and the algorithm gets more confident about which portions of the parameter space are worth examining and which are not. Gradient-based optimization methods [26] are often used in neural network models, where the gradient of the hyperparameters is mainly calculated and optimized by a gradient descent algorithm. The idea of evolutionary optimization methods [27] comes from biological concepts, and since natural evolution is a dynamic process occurring in a constantly changing environment, it applies to the hyperparameter search problem, since hyperparameter search is also a dynamic process.

3. Method

3.1. Frequency-Based Data Mix. Convolutional neural networks (CNNs) have shown excellent performance in computer vision tasks, particularly in classification tasks. To increase robustness in real-world scenarios, CNNs often adopt two practical strategies: data augmentation and model integration. Data augmentation reduces overfitting and improves the generalization of the model. Traditional image enhancement is label preserving, e.g., flipping and cropping. Multiple inputs and their labels are proportionally mixed to create artificial samples in mixed-sample data augmentation (MSDA) approaches [28]. MSDA is simple to implement and really helps to improve performance, so it is widely used in areas such as image recognition, sound recognition, GAN, and semisupervised learning.

Recently, hybrid-based enhancement techniques have achieved highly accurate prediction performance. They generate input data by a linear combination of two randomly selected training data. Likewise, their corresponding labels are generated by the same linear combination of two labels. By doing so, they effectively improve prediction accuracy while preventing some undesirable behaviors such as memory and sensitivity to adversarial examples. Moreover, Mixup training encourages the output of the DNN [29], i.e., the estimated label distribution, as a better indicator of the actual likelihood of correcting the predictions. Specifically, for generating an enhanced sample, the calculation for Mixup training is as follows

$$x_{\text{mix}} = \begin{cases} \lambda x_1 + (1-\lambda)x_2, \\ \mathrm{M} \odot x_1 + (1-\mathrm{M}) \odot x_2, \end{cases} \quad (1)$$
$$y_{\text{mix}} = \lambda y_1 + (1-\lambda)y_2, \lambda \sim \text{Beta}(\alpha, \alpha),$$

where x and y denote a training sample and its label, $M \in \{0,1\}^{W \times H}$ denotes a binary mask indicating the position of the reject and fill from the two images, $\odot$ is an element multiplication, Beta$(\bullet, \bullet)$ implies the beta distribution, and $\alpha \in (0,\infty)$ is the parameter controlling the shape of the beta distribution. Using mixed inputs and mixed labels, the model minimizes the following equation.

$$\mathscr{L}_{\text{mix}} = \lambda \mathscr{H}\left(\tilde{y}_{\text{mix}}, y_1\right) + (1-\lambda)\mathscr{H}\left(\tilde{y}_{\text{mix}}, y_2\right), \quad (2)$$

where $\tilde{y}(=\sigma(f(x)))$ denotes the predicted label distribution from the model, f is the model, σ is the activation function, usually a softmax function, $\sigma_{\text{sm}}(z) = \exp(z)/\sum_{i=1}^{N} \exp(z_i)$, and $\mathscr{H}$ is the cross-entropy function formulated by $\mathscr{H}(p,q) = -\int_x p(x) \log q(x)$.

Because the Mixup's raw picture exploration is restricted, the suggested FreMix performs a fast Fourier transform (FFT) on the input image first, extracting the frequency information rather than the original pixel information for subsequent augmentation. Figure 1 depicts the flow of our technique. The fast Fourier transform (FFT) is a technique for computing a sequence's discrete Fourier transform (DFT) or inverse transform fast [30]. Fourier analysis converts a signal's original domain (typically time or space) into a frequency domain representation or vice versa. By reducing the DFT matrix into the product of sparse (mainly zero) elements, the FFT can quickly perform such a transform.

The DFT requires the computation of approximately N^2 multiplications and N^2 additions. This computation is large when N is large. The N-point DFT is decomposed into two $N/2$-point DFTs using the symmetry and periodicity of W_N; therefore, the total computation of two $N/2$-point DFTs is just half of the original, i.e., $(N/2)^2 + (N/2)^2 = N^2/2$. This can be continued by decomposing $N/2$ into $N/4$-point DFTs again, etc. For $N = 2\,\mathrm{m}$, points of DFT can be decomposed into 2 points of DFT, so that its computation can be reduced to $(N/2)\log_2 N$ times multiplication and $Nlog2N$ times addition. Here are the steps of the operation.

The DFT of a finite length discrete signal $x(n), n = 0, 1, \cdots, N-1$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \mathrm{k} = 0, 1, \cdots, \mathrm{N}-1, W_N = e^{-j\frac{2\pi}{N}}. \quad (3)$$

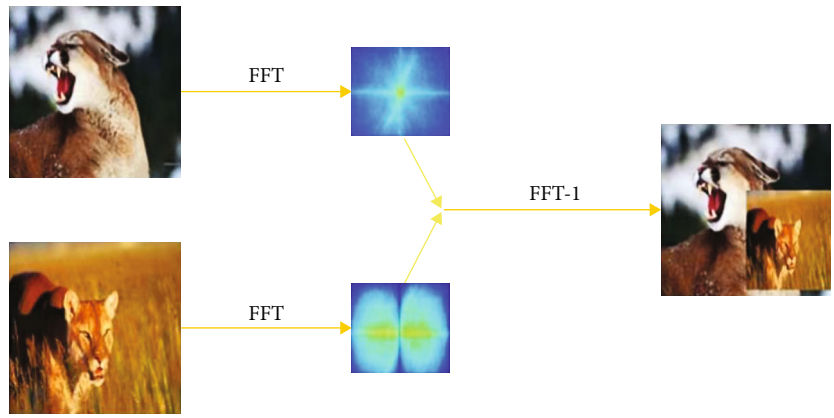

Figure 1: Schematic diagram of our approach.

Decomposing $x(n)$ into the sum of two sequences of even and odd numbers, i.e.,

$$x(n) = x_1(n) + x_2(n), \tag{4}$$

where $x_1(n)$ and $x_2(n)$ are both of length $N/2$, $x_1(n)$ is an even sequence, and $x_2(n)$ is an odd sequence, then

$$X(k) = \sum_{n=0}^{(N/2)-1} x_1(n) W_N^{2kn} + \sum_{n=0}^{N/2} x_2(n) W_N^{(2n+1)k} (\mathrm{k} = 0, 1, \cdots, \mathrm{N} - 1). \tag{5}$$

Because $W_N^{2kn} = e^{-j(2\pi/N)2kn} = e^{-j(2\pi/N/2)kn} = W_{N/2}^{kn}$, then

$$\begin{aligned} X(k) &= \sum_{n=0}^{(N/2)-1} x_1(n) W_{N/2}^{kn} + W_N^k \sum_{n=0}^{(N/2)-1} x_2(n) W_{N/2}^{kn} \\ &= X_1(k) + W_N^k X_2(k) \ (k = 0, 1, \cdots, N-1), \end{aligned} \tag{6}$$

where $X_1(k)$ and $X_2(k)$ are the $N/2$-point DFTs of $x_1(n)$ and $x_2(n)$, respectively. Since both $X_1(k)$ and $X_2(k)$ have period $N/2$ and $W_N^k + N/2 = -W_N^k$, $X(k)$ can again be expressed as

$$\begin{aligned} X(k) &= X_1(k) + W_N^k X_2(k) \left(k = 0, 1, \cdots, \frac{N}{2} - 1\right), \\ X\left(k + \frac{N}{2}\right) &= X_1(k) - W_N^k X_2(k) \ (k = 0, 1, \cdots, N/2 - 1). \end{aligned} \tag{7}$$

The principle of the FFT algorithm is to achieve large-scale transformations by many small more easily performed transformations, reducing the operational requirements and increasing the speed with the operation. The FFT is not an approximation of the DFT, they are exactly equivalent. Figure 2 shows the flow chart of 8-point FFT decomposition.

3.2. Adaptive Fusion Algorithm. FreMix creates new samples by linearly interpolating pairs of FFT information of raw examples, and it is simple to do and to be effective in picture classification problems. There are two issues. First, FreMix asks for more epochs to converge. The reason is that it needs more extended training and explores more regions of the data space. Second, FreMix has a condition of hyperparameter value to sample mixing coefficients, while various hyperparameter values usually cause large differences in model accuracy. To mitigate this problem, inspired by Mixup without hesitation (mWh) [31], we integrate the exploration-exploitation mechanism from reinforcement learning into our FreMix to overcome both problems, rather than enriching data by using Mixup during the model training process. FreMix will be speeded up since exploration-exploitation will turn the mixing operation off. And it also makes it robust to the hyperparameter α. Note that mWh is a pixel domain-based method, which has the disadvantage of requiring extraction from the original pixels compared to the frequency domain. We utilize the exploration-exploitation (EE) mechanism to operate in the frequency domain in the proposed FreMix. Our method can extract frequency information directly instead of the original pixels for further data augmentation.

Specifically, we propose an adaptive fusion algorithm, which assumes that the number of minibatches is m during training and defines two parameters p and q ($0 \le p < q \le 10 \le p < q \le 1$), which divide the training process into three phases:

Step 1. From 1 to pm minibatch, train using Mixup.

Step 2. From pm + 1pm + 1 to qm, switch between Mixup and base data enhancement algorithms.

Step 3. Run Mixup with probability ϵ, where ϵ decreases linearly from 1 to 0.

In Step 1, the FreMix algorithm uses Mixup to search in the large percentage of the sample representation area. In Step 2, there is a trade-off between exploration and exploitation. Step 3 gradually switches from the exploration model to the exploitation model. As for effective training, FreMix is a general and straightforward training policy. To balance exploration and exploitation, we use the method of reintroducing essential data augmentation. When comparing

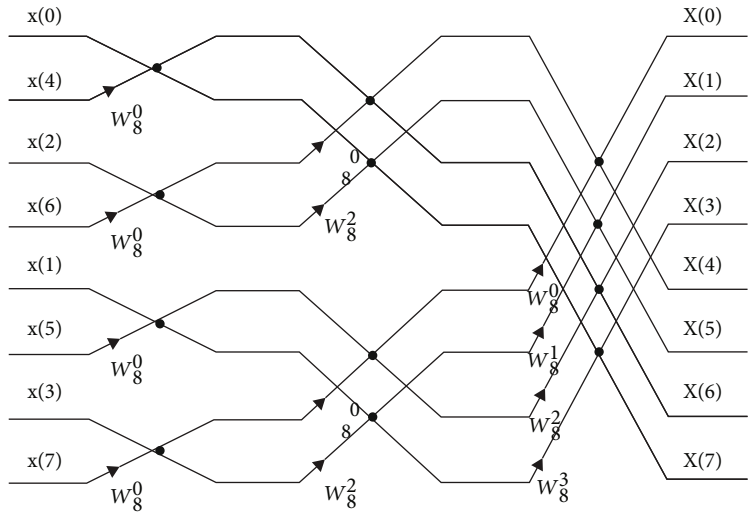

Figure 2: Illustration of an 8-point FFT decomposition procedure.

CutMix and mWh in CutMix with ResNet-50, experimental results showed that the convergence rate of diverse dataset instances is improved by mWh and the accuracy on ImageNet is robust. Compared with the baseline, it also gets significant improvement in various tasks and models. Our FreMix method based on mWh in the frequency domain gives further improved results.

4. Experiments

We evaluate our method in a variety of tasks and compare it to the competition. We emphasize that our approach can be combined with any model structure to enjoy its performance benefits. This is simply done by adding our data augmentation technique before training the model.

4.1. Model Architecture. We choose five CNN architectures as backbone networks: three of them are traditional CNN (i.e., VGGNet, ResNet, and ResNeXt) [32] and the others are lightweight CNN (i.e., MobileNetV2 and ShuffleNet) [33] datasets. We validated the effectiveness of our method on three benchmark datasets ranging from small to large: CIFAR100 (32×32 RGB images, 100 classes), Tiny imagenet (64×64 RGB images, 100 classes) and ILSVRC2015 (256×256 RGB images, 1000 classes).

4.2. Evaluation Settings. The best training method for all networks was stochastic gradient decay with a momentum of 0.9. All methods used for our comparisons followed the same training schedule and dataset. For CIFAR100, we set the initial learning rate to 0.1 and decayed the learning rate by 0.2 every 60, 120, 160, and 200 epochs. In Tiny imagenet and ILSVRC2015, we set the initial learning rate to 0.1 and decayed it by 0.1 in 75, 150, and 225 calendars. Because lightweight models have a different ideal training scheme, we followed the procedure described in their paper. We used a weight decay of 4e-5 for CIFAR100 and 1e-4 for the other models to regularise the model. For the CIFAR100, Tiny imagenet, and ILSVRC datasets, the batch sizes for each model are 256.

4.3. Classification Accuracy and Expected Calibration Error. We will show that our approach improves classification accuracy (i.e., high confidence on test samples). Besides, in this subsection section, we experiment with a more realistic scenario. For the quantitative analysis of confidence calibration, we use two popular metrics, expected calibration error (ECE) and excess confidence error (OE).

The ECE represents the average difference between true confidence and predicted confidence. If ECE is zero, it means that the network is correctly calibrated. OE is similar to ECE, but it only measures the difference in confidence when overconfidence is indicated. Overconfidence is primarily a critical factor in high-risk systems. Therefore, this metric is a good indicator to assess system reliability for high-risk applications. These two metrics were calculated on the validation set.

The experimental findings employing various networks and data sets are shown in Tables 1, 2, and 3. After combining our methods, we consistently achieved better accuracy and confidence calibration. In particular, our gains in prediction accuracy are substantial, with the gap between the baseline and the baseline combined with our method being as large as the gap between vanilla and other competitors. As a result, our method exceeds the performance of existing methods under most experimental conditions. A key observation can be made by experimenting with compact models like MobileNetV2. In general, a Mixup-like approach is an enhancement method that populates training examples to prevent overfitting. However, if it injects examples that are far from the training distribution, such an augmentation method can cause underfitting. Underfitting usually does not degrade the performance of high-volume networks, but it can impair the performance of low-volume networks. Since MobileNetV2 is a low-volume model, it requires less regularization than larger models, and applying weak regularization (i.e., small weight decay) may be sufficient. When CutMix

Table 1: The results of our experiments on CIFAR100.

Network	Metric	Vanilla	Mixup	FreMix	CutMix	FreMix+EE
VGG16	Acc	74.30	75.02	76.22	75.34	76.10
	ECE	0.18	0.06	0.03	0.06	0.06
	OE	0.16	0.04	0.02	0.03	0.08
ResNet50	Acc	78.32	79.82	80.96	80.57	81.02
	ECE	0.09	0.04	0.02	0.08	0.07
	OE	0.07	0.03	0.01	0.06	0.06
ResNeXt50	Acc	79.18	81.10	81.63	81.16	81.46
	ECE	0.06	0.04	0.02	0.059	0.03
	OE	0.05	0.01	0.00	0.047	0.02
MobileNetV2	Acc	69.69	69.96	73.90	68.82	69.91
	ECE	0.06	0.01	0.04	0.05	0.04
	OE	0.04	0.01	0.01	0.01	0.00
ShuffleNetV2	Acc	72.17	74.17	75.53	73.60	73.73
	ECE	0.08	0.06	0.042	0.01	0.02
	OE	0.06	0.00	0.000	0.01	0.00

Table 2: The results of our experiments on Tiny imagenet.

Network	Metric	Vanilla	Mixup	FreMix	CutMix	FreMix+EE
ResNet50	Acc	66.6	68.34	70.71	69.08	69.87
	ECE	0.09	0.032	0.03	0.029	0.03
	OE	0.07	0.022	0.01	0.015	0.05
MobileNetV2	Acc	57.62	59.55	62.12	53.54	57.66
	ECE	0.08	0.09	0.03	0.09	0.08
	OE	0.05	0.02	0.00	0.00	0.00

Table 3: The results of our experiments on ILSVRC2015.

Network	Metric	Vanilla	Mixup	FreMix	CutMix	FreMix+EE
ResNet50	Acc	76.13	77.37	78.38	78.43	78.51
	ECE	0.37	0.04	0.03	0.03	0.02
	OE	0.03	0.01	0.01	0.03	0.03

was used to train MobileNetV2 on Tiny imagenet, we observed severe performance degradation, and we speculate that the accuracy degradation is due to the strong regularization induced underfitting.

In contrast, when our method is combined with CutMix, the impact of this underfitting is greatly reduced, filling the degradation gap caused by CutMix. From this result, we believe that our method does not penalize vanilla training to prevent overfitting. Instead, our method helps the model understand the hidden relationships by weak supervision. Since hidden relationships can provide a reasonable explanation for understanding examples far from the training distribution, our method can also prevent underfitting. This observation is consistent with our motivation that interclass correlations help improve prediction accuracy and prediction confidence estimates.

Table 4: The difference between our method and other methods.

Method	Coefficient	Fusion	Training
Mixup	Manual	Pixel field	Fixed
CutMix	Manual	Pixel field	Fixed
Ours	Automatic	Frequency field	Adaptive

5. Discussion

5.1. The Difference between Our Method and Other Methods. The coefficients of Mixup need to be adjusted manually. They are fused in the pixel domain. The coefficients of CutMix also need to be adjusted manually, and the trained model is fixed for both. Mixing in the pixel domain affects the pixel-based image properties. Mixup injects examples that are far from the training distribution, leading to the underfitting of the data augmentation. CutMix generally leads to underfit problems and may lead to confusion in model classification. In our method, the coefficients can be adjusted automatically within a preset range (for a comparison of Mixup and CutMix with our method in terms of coefficients, fusion domain, and training patterns, see the Table 4 for details).

5.2. Shortcomings and Future Work. Although our method mines the image data and expands the experimental data

in the field of data augmentation, it will add extra computation. Another point is that although our method can automatically adjust the coefficients, the range of adjustment needs to be set. Then, the parameters are adjusted within the range we set. That means the adaptive adjustment of the parameters is within a certain range that we set manually. It is not completely free of manual operation. Our future research direction is to add more automatic adjustment settings to avoid manual hyperparameter adjustment as much as possible.

6. Conclusion

Generally, existing deep learning models rely on a great number of data. Data augmentation has a strong generalization ability to raise the training data set as well as make the data set as diverse as possible. The current class of augmentation methods based on mixing different samples, including Mixup and CutMix, can be very effective in improving the model's accuracy. However, since these methods operate mainly in the pixel domain, they cannot process successfully mined and fused image data, and the fused weighting factors are mainly modified manually, so they are not suitable for practical applications.

To better exploit the frequency information for data augmentation and improve the existing sample-based fusion strategies, we propose a novel FreMix method that fuses with extracted frequencies and performs adaptive exploration of the fused coefficients. Our method achieves very good performance on several datasets with different model structures, such as CIFAR and ImageNet, which greatly surpasses previous methods. Further, despite the effectiveness, it should be noted that the proposed method increases the computational effort and the coefficients of adaptive exploration fusion are not fully automatic, and a range needs to be set manually. Our future work will focus on improving the efficiency of frequency fusion as well as adding more automatic adjustment settings to avoid manual adjustment of hyperparameters.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Yang Xiu and Xinyi Zheng conceptualized the study; Yang Xiu and Xinyi Zheng were responsible for the methodology; Yang Xiu was responsible for the project administration; Linlin Sun validated the study; Linlin Sun and Zhuohao Fang visualized the study; Yang Xiu wrote the original draft; Linlin Sun and Zhuohao Fang wrote, reviewed, and edited the manuscript.

References

[1] P. Skalski, "Preventing deep neural network from overfitting," *Towards Data Science*, vol. 7, 2018.

[2] D. Su, H. Kong, Y. Qiao, and S. Sukkarieh, "Data augmentation for deep learning based semantic segmentation and crop-weed classification in agricultural robotics," *Computers and Electronics in Agriculture*, vol. 190, pp. 106418–106418, 2021.

[3] X. Bai, X. Wang, X. Liu et al., "Explainable deep learning for efficient and robust pattern recognition: a survey of recent developments," *Pattern Recognition*, vol. 120, article 108102, 2021.

[4] W. Sirichotedumrong, T. Maekawa, Y. Kinoshita, and H. Kiya, "Privacy-preserving deep neural networks with pixel-based image encryption considering data augmentation in the encrypted domain," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 674–678, IEEE, Taipei, Taiwan, 2019.

[5] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *2017 Chinese automation congress (CAC)*, pp. 4165–4170, IEEE, 2017.

[6] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 4, pp. 593–597, 2019.

[7] J. Cui, X. Zhang, F. Xiong, and C.-L. Chen, "Pathological myopia image recognition strategy based on data augmentation and model fusion," *Journal of Healthcare Engineering*, vol. 2021, Article ID 5549779, 15 pages, 2021.

[8] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[9] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," https://arxiv.org/abs/1710.09412.

[10] H. Zhang and J. Wang, "Defense against adversarial attacks using feature scattering-based adversarial training," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] J. C. White, M. A. Wulder, G. W. Hobart et al., "Pixel-based image compositing for large-area dense time series applications and science," *Canadian Journal of Remote Sensing*, vol. 40, no. 3, pp. 192–212, 2014.

[12] J. H. Lee, M. Z. Zaheer, M. Astrid, and S. I. Lee, "Smoothmix: a simple yet effective data augmentation to train robust classifiers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 756-757, Seattle, WA , USA., 2020.

[13] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," https://arxiv.org/abs/1708.04552.

[14] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, Korea, 2019.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6023–6032, Salt Lake City, USA., 2018.

[16] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, San Francisco, California, USA., 2017.

[18] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Frontiers in Neuroscience*, vol. 13, p. 95, 2019.

[19] S. Zagoruyko and N. Komodakis, "Wide residual networks," https://arxiv.org/abs/1605.07146.

[20] H. Guo, Y. Mao, and R. Zhang, "Mixup as locally linear out-of-manifold regularization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 3714–3722, 2019.

[21] V. Verma, A. Lamb, C. Beckham et al., "Manifold mixup: better representations by interpolating hidden states," *International Conference on Machine Learning*, vol. 97, 2019.

[22] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of imagenet as an alternative to the cifar datasets," https://arxiv.org/abs/1707.08819.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[24] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: a big comparison for NAS," https://arxiv.org/abs/1912.06059.

[25] I. Dewancker, M. McCourt, S. Clark, P. Hayes, A. Johnson, and G. Ke, "A stratified analysis of Bayesian optimization methods," https://arxiv.org/abs/1603.09441.

[26] A. Jameson, "Gradient based optimization methods," *MAE Technical Report*, p. 2057, 1995.

[27] A. J. Keane, "A brief comparison of some evolutionary optimization methods," in *Modern Heuristic Search Methods*, pp. 255–272, John Wiley, Chichester, UK, 1996.

[28] A. Ramé, R. Sun, and M. Cord, "Mixmo: mixing multiple inputs for multiple outputs via deep subnetworks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 823–833, 2021.

[29] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[30] P. Heckbert, "Fourier transforms and the fast Fourier transform (FFT) algorithm," *Computer Graphics*, vol. 2, pp. 15–463, 1995.

[31] H. Yu, H. Wang, and J. Wu, "Mixup without hesitation," in *International Conference on Image and Graphics*, pp. 143–154, Springer, Cham, 2021.

[32] G. Pant, D. P. Yadav, and A. Gaur, "ResNeXt convolution neural network topology-based deep learning model for identification and classification of *Pediastrum*," *Algal Research*, vol. 48, p. 101932, 2020.

[33] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, Munich, Germany., 2018.