ChimeraMix: Image Classification on Small Datasets via Masked Feature Mixing

Christoph Reinders[*] , **Frederik Schubert**[*] and **Bodo Rosenhahn**

Institute for Information Processing, Leibniz University Hannover

{reinders,schubert,rosenhahn}@tnt.uni-hannover.de

Abstract

Deep convolutional neural networks require large amounts of labeled data samples. For many real-world applications, this is a major limitation which is commonly treated by augmentation methods. In this work, we address the problem of learning deep neural networks on small datasets. Our proposed architecture called *ChimeraMix* learns a data augmentation by generating compositions of instances. The generative model encodes images in pairs, combines the features guided by a mask, and creates new samples. For evaluation, all methods are trained from scratch without any additional data. Several experiments on benchmark datasets, e.g., ciFAIR-10, STL-10, and ciFAIR-100, demonstrate the superior performance of *ChimeraMix* compared to current state-of-the-art methods for classification on small datasets. Code is available at https://github.com/creinders/ChimeraMix.

1 Introduction

Large-scale datasets contribute significantly to the success of deep neural networks in computer vision and machine learning in recent years. The collection of massive amounts of labeled data samples, however, is very time-consuming and expensive. Less explored is the research direction of applying deep learning algorithms on small data problems. These small data problems are common in the real world. In many applications, there is not much data available or cannot be used due to legal reasons [Renard *et al.*, 2020].

When learning with a limited amount of data, most research focuses on transfer learning, self-supervised learning, and few-shot learning techniques. Transfer [Neyshabur *et al.*, 2020] and self-supervised methods [Assran *et al.*, 2021] generate a representation on a large source dataset and transfer the knowledge to the target domain where the model can be fine-tuned. Similarly, few-shot learning methods [Kolesnikov *et al.*, 2020] are trained on a base set to generalize to a novel set given a small number of support examples. All these approaches, however, require a large source dataset of annotated data samples and the source domain needs to be close to the

---

[*]Equal contribution
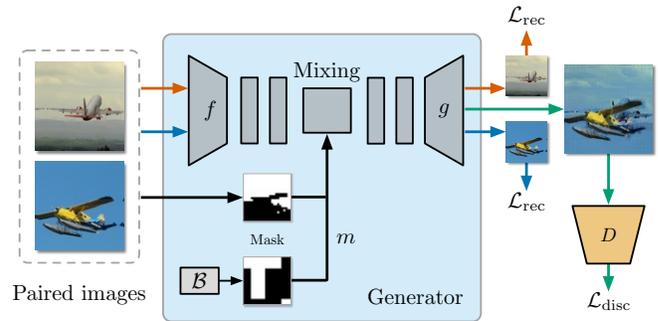
Figure 1: **Overview of ChimeraMix** – Two images of the same class are passed through a feature extractor f. Their features are mixed according to a mask m and the generator-discriminator architecture learns to generate new samples. Additionally, a reconstruction loss $\mathcal{L}_{\text{rec}}$ improves the sample quality and stabilizes the training process of ChimeraMix. The mask is either sampled from a binomial distribution $\mathcal{B}$ or computed using a segmentation algorithm.

target domain. Additionally, different input sensors (e.g., hyperspectral camera or depth sensor) or copyright conditions often prevent the use of readily available datasets.

In this work, we address the challenging small data problem by presenting a novel feature mixing architecture for generating images. Our method makes use of the fact that the label of the classification task is invariant under the composition of object instances. Thus, given multiple instances of a class our generative model is able to compose new samples guided by a mask. We present two methods to generate the mixing masks, one based on a grid of rectangular patches and one that uses a segmentation algorithm.

The generative process and training of our method *ChimeraMix* is outlined in Fig. 1. Overall, we construct our method as a generative-adversarial architecture. The masks are used to mix the features at different locations in the images. Subsequently, the generator network decodes and refines the features to create new compositions (examples are shown in Fig. 2). We evaluate our method on benchmarks and raise the state-of-the-art accuracy in the small data setting on ciFAIR-10, STL-10, and ciFAIR-100. Additionally, we provide in-depth analyses to demonstrate the impact of the generator and combine our method with different automatic augmentation methods to improve the accuracy even further.

To summarize, our **contributions** are:

- We propose a novel generative approach for addressing the small data image classification task.

- Our generator introduces a feature mixing architecture. Guided by a mask, the generator learns to combine images and create new compositions of the instances.

- Experiments on benchmark datasets demonstrate that ChimeraMix outperforms current state-of-the-art methods in small data image classification.

- Our experiments show that ChimeraMix can be combined with other augmentation methods to improve their performance even further.

2 Related Work

Learning deep neural network from limited amount of labeled data has been studied from various perspectives. Many methods have been proposed in the field of transfer learning and few-shot learning. Small data learning differs from both research areas since all networks are trained from scratch and only a small number of labeled training examples is used without any additional data.

Deep neural networks have thousands or millions of parameters that need to be trained. To avoid overfitting, various regularization techniques have been presented such as Dropout [Srivastava *et al.*, 2014] or BatchNormalization [Ioffe and Szegedy, 2015]. However, when training on small datasets these regularization mechanisms are not able to prevent overfitting and other ways of regularization are required.

Data Augmentation. Data augmentation increases the size of the dataset by applying random transformations to the original data samples generating new synthetic training samples. Standard transformations for image classification are random cropping of patches and horizontal or vertical flipping [Krizhevsky *et al.*, 2012]. Cutout [DeVries and Taylor, 2017] masks a square region in the image to improve the robustness of the networks. Similarly, Random Erasing [Zhong *et al.*, 2020] replaces a region in the image with random values. Automatic augmentation methods for learning the augmentation strategy have been proposed in recent years combining multiple geometric and color transformations. AutoAugment [Cubuk *et al.*, 2019] optimizes the parameters of the policies using a recurrent neural network via reinforcement learning. Since this process is computationally expensive and requires training data, Müller and Hutter [2021] propose TrivialAugment which is parameter-free. The augmentation method randomly samples a single policy per image that is applied with a uniformly sampled strength.

The performance of different network architectures in relation to the amount of training data has been analyzed by Brigato and Iocchi [2021]. Simple models achieve good results when little data is available, however, deeper networks catch up when augmentation is applied. The authors suggest that further data generation and augmentation methods could further boost the performance. Bornschein *et al.* [2020] analyze the generalization performance of deep neural networks depending on the size of the dataset empirically and found
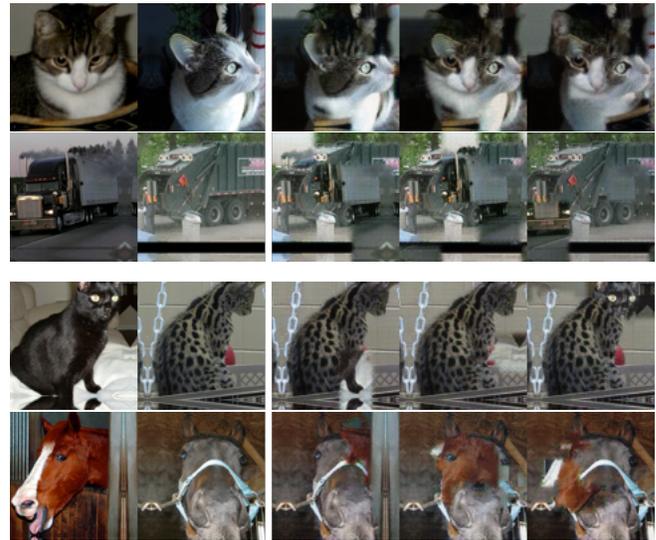
Figure 2: Examples of generated images by ChimeraMix+Grid (top) and ChimeraMix+Seg (bottom) on STL-10. ChimeraMix combines image pairs (first two columns), mixes the features guided by a sampled mask and generates new image compositions (last three columns).

that large networks mostly outperform small networks even on small datasets. Barz and Denzler [2020a] propose the use of the cosine loss function instead of the cross-entropy loss, or a combination of both, in the small data regime. While the cross-entropy loss moves the activations towards infinity, the cosine loss function includes an l^2 normalization as a regularizer. Arora *et al.* [2020] explore the performance of convolutional neural tangent kernels (CNTK) compared to a ResNet trained on small amount of data. The combination of convolutional neural networks for feature learning and random forests as robust classifiers has been demonstrated in [Reinders *et al.*, 2018; Reinders *et al.*, 2019]. Gauthier *et al.* [2021] present a parametric scattering transform to learn descriptive representations. The authors introduce a differentiable architecture that learns the parameters of the Morlet wavelet filters.

Mixing Augmentation. While classic data augmentation processes one image at a time, there are several approaches that use multiple samples. MixUp [Zhang *et al.*, 2018] generates weighted combinations of random image pairs by linear interpolation between the images and targets. CutMix [Yun *et al.*, 2019] replaces a rectangular region of one image with the content from another image. Generative approaches for synthesizing new samples have been used in a variety of domains, such as video game levels [Awiszus *et al.*, 2020; Schubert *et al.*, 2021] and object detection [Kluger *et al.*, 2018]. Generative Latent Implicit Conditional Optimization (GLICO) [Azuri and Weinshall, 2021] is a generative method for synthesizing new samples using spherical interpolation in the latent space. The method has a learnable latent representation for each training sample and optimizes the reconstruction by the generator. An additional classification loss tunes the semantic structure of the latent space. Finally, a method that also uses masks for data augmentation is SuperMix [Dabouei

Samples per Class		5	10	20	30	50	100
Dataset	Method						
ciFAIR-10	Baseline	31.37 ± 3.28	38.09 ± 1.34	47.50 ± 2.09	53.19 ± 0.60	58.84 ± 0.82	70.34 ± 1.17
	Cutout	28.88 ± 2.84	37.33 ± 1.03	47.55 ± 2.06	53.39 ± 1.32	61.17 ± 1.03	72.14 ± 1.10
	Random Erasing	28.91 ± 2.64	37.13 ± 0.61	47.20 ± 2.32	53.11 ± 1.65	60.34 ± 0.35	72.00 ± 0.71
	Cosine	31.45 ± 3.22	37.88 ± 1.24	46.69 ± 1.38	52.16 ± 0.72	59.24 ± 1.60	70.18 ± 1.32
	MixUp	33.41 ± 2.70	43.03 ± 1.21	53.09 ± 1.00	59.47 ± 1.10	66.16 ± 0.78	74.23 ± 0.35
	Scattering	30.50 ± 3.87	37.28 ± 1.87	45.65 ± 1.45	50.47 ± 1.19	54.30 ± 0.95	61.51 ± 0.79
	GLICO	31.91 ± 2.41	42.02 ± 0.87	51.61 ± 1.23	59.03 ± 0.70	65.00 ± 1.24	73.96 ± 0.81
	ChimeraMix+Grid	36.94 ± 2.63	45.57 ± 2.11	53.67 ± 2.84	59.66 ± 1.35	65.42 ± 0.83	73.76 ± 0.30
	ChimeraMix+Seg	$\mathbf{37.31 \pm 2.57}$	$\mathbf{47.60 \pm 1.81}$	$\mathbf{56.21 \pm 1.77}$	$\mathbf{60.92 \pm 0.62}$	$\mathbf{67.30 \pm 1.21}$	$\mathbf{74.96 \pm 0.21}$
STL-10	Baseline	27.61 ± 0.90	31.93 ± 1.68	36.50 ± 0.94	39.95 ± 1.26	44.82 ± 0.48	53.51 ± 1.65
	Cutout	28.05 ± 1.73	31.45 ± 2.46	37.68 ± 1.30	40.69 ± 1.13	45.63 ± 1.19	54.32 ± 1.01
	Random Erasing	27.87 ± 1.36	31.32 ± 0.48	36.91 ± 1.45	40.66 ± 0.84	45.93 ± 1.10	53.31 ± 1.52
	Cosine	25.97 ± 0.93	30.37 ± 1.34	35.51 ± 0.95	40.05 ± 1.01	45.51 ± 1.23	53.01 ± 1.09
	MixUp	30.06 ± 1.80	35.63 ± 0.85	42.44 ± 1.85	45.00 ± 2.71	49.03 ± 1.34	54.38 ± 2.11
	GLICO	26.97 ± 0.98	33.02 ± 1.07	37.88 ± 1.22	42.66 ± 0.66	48.40 ± 0.72	54.82 ± 1.94
	ChimeraMix+Grid	$\mathbf{32.18 \pm 0.90}$	37.01 ± 0.84	43.19 ± 1.03	48.93 ± 1.34	52.81 ± 1.45	60.04 ± 0.27
	ChimeraMix+Seg	31.37 ± 1.72	$\mathbf{37.05 \pm 1.09}$	$\mathbf{44.74 \pm 0.60}$	$\mathbf{49.58 \pm 0.49}$	$\mathbf{55.06 \pm 1.11}$	$\mathbf{60.44 \pm 0.71}$
ciFAIR-100	Baseline	18.78 ± 0.79	24.53 ± 0.28	39.27 ± 0.31	45.99 ± 0.32	53.40 ± 0.36	61.81 ± 0.41
	Cutout	19.25 ± 0.52	27.77 ± 0.39	40.72 ± 0.68	47.78 ± 0.39	55.13 ± 0.30	$\mathbf{63.26 \pm 0.62}$
	Random Erasing	18.35 ± 0.37	26.09 ± 0.74	38.83 ± 1.01	46.14 ± 0.38	54.26 ± 0.08	63.24 ± 0.50
	Cosine	18.04 ± 0.87	23.72 ± 0.35	38.84 ± 0.73	45.83 ± 0.43	53.32 ± 0.11	61.50 ± 0.46
	MixUp	20.63 ± 0.16	31.03 ± 0.54	41.58 ± 0.40	47.88 ± 0.45	54.87 ± 0.20	62.49 ± 0.52
	Scattering	12.67 ± 0.40	18.25 ± 0.56	26.37 ± 0.63	31.51 ± 0.28	36.49 ± 0.42	48.18 ± 0.33
	GLICO	19.32 ± 0.39	28.49 ± 0.60	40.45 ± 0.30	45.90 ± 0.77	53.53 ± 0.19	60.68 ± 0.50
	SuperMix	19.23 ± 0.45	26.78 ± 0.20	38.47 ± 0.83	44.69 ± 0.63	53.07 ± 0.13	62.63 ± 0.30
	ChimeraMix+Grid	20.24 ± 0.12	31.62 ± 0.82	41.80 ± 0.52	48.10 ± 0.71	54.67 ± 1.01	62.13 ± 0.27
	ChimeraMix+Seg	$\mathbf{21.09 \pm 0.47}$	$\mathbf{32.72 \pm 0.60}$	$\mathbf{43.23 \pm 0.38}$	$\mathbf{48.83 \pm 0.72}$	$\mathbf{55.79 \pm 0.21}$	62.96 ± 0.77

Table 1: Test accuracy on ciFAIR-10, STL-10, and ciFAIR-100. Cutout [DeVries and Taylor, 2017], Random Erasing [Zhong *et al.*, 2020], Cosine [Barz and Denzler, 2020a], MixUp [Zhang *et al.*, 2018], Scattering [Gauthier *et al.*, 2021], GLICO [Azuri and Weinshall, 2021], and SuperMix [Dabouei *et al.*, 2021] are state-of-the-art methods for small data image classification. A standard classification is included as a baseline. The best result is highlighted in bold. Note, that the size of the dataset of ciFAIR-100 with 5 samples per class is the same as that of ciFAIR-10 and STL-10 with 50 samples.

et al., 2021]. SuperMix introduces a student-teacher architecture to optimize the masks according to which the two images are mixed.

3 Method

In this work, we present ChimeraMix, a novel method for learning deep neural networks with a small amount of data. ChimeraMix is a generative approach that learns to create new compositions of image pairs from the same class by mixing their features guided by a mask.

3.1 Mixing Generator

ChimeraMix's general architecture is shown in Fig. 1. Given a labeled image dataset, in each iteration a pair of images $x_1, x_2 \in \mathbb{R}^{H \times W \times C}$ from the same class is sampled, where $H \times W$ is the image size and C the number of color channels. ChimeraMix consists of an encoder and a decoder module. The encoder f extracts features $e_i = f(x_i) \in \mathbb{R}^{H' \times W' \times C'}$ with a downsampled size of $H' \times W'$ and C' dimensions. At each spatial location, a mask $m \in \{0, 1\}^{H' \times W'}$ selects the

feature vector from one of both images. Finally, the decoder g generates new image compositions $\hat{x}$ based on the mixed features, i.e.,

$$\hat{x} = g\left(e_1 \odot m + e_2 \odot (1 - m)\right). \quad (1)$$

Training. To generate valid compositions, we introduce a discriminator D that tries to distinguish real and generated images performing a binary classification. During training, the discriminator is optimized by minimizing the mean squared error $\mathcal{L}_{D,disc}$ between the predicted label and actual label. The generator, on the other hand, learns to create realistic image compositions by minimizing the mean squared error between the discriminator's prediction of the generated image and the real image class (denoted as $\mathcal{L}_{G,disc}$), i.e., the discriminator cannot identify if the image is real or generated. The discriminator and generator are trained alternately. Additionally, the generator is guided by reconstructing the original images. When the mask m is set to zeros, the image x_1 should be reconstructed, and vice versa for x_2 when m consists of ones. The reconstruction loss minimizes the similarity between the generator's output and the corresponding

| Samples per Class | | | 5 | 10 | 20 | 30 | 50 | 100 |
Method	AA	TA						
AutoAugment	✔	–	21.39 ± 0.95	29.56 ± 0.68	40.81 ± 0.35	47.58 ± 0.56	55.01 ± 0.24	63.69 ± 0.42
TrivialAugment	–	✔	23.85 ± 0.60	32.48 ± 0.34	44.13 ± 0.16	50.17 ± 0.26	56.27 ± 0.19	64.02 ± 0.18
ChimeraMix+Grid	–	–	20.24 ± 0.12	31.62 ± 0.82	41.80 ± 0.52	48.10 ± 0.71	54.67 ± 1.01	62.13 ± 0.27
ChimeraMix+Grid	✔	–	25.24 ± 1.02	34.60 ± 0.47	45.16 ± 0.38	51.00 ± 0.87	57.74 ± 0.51	64.19 ± 0.68
ChimeraMix+Grid	–	✔	25.69 ± 0.37	34.67 ± 0.51	45.78 ± 0.10	51.81 ± 0.11	57.80 ± 0.62	64.21 ± 0.37
ChimeraMix+Seg	–	–	21.09 ± 0.47	32.72 ± 0.60	43.23 ± 0.38	48.83 ± 0.72	55.79 ± 0.21	62.96 ± 0.77
ChimeraMix+Seg	✔	–	25.16 ± 0.37	35.02 ± 0.55	45.27 ± 0.09	51.25 ± 0.67	57.86 ± 0.41	64.39 ± 0.43
ChimeraMix+Seg	–	✔	$\mathbf{26.36 \pm 0.17}$	$\mathbf{36.02 \pm 0.22}$	$\mathbf{46.61 \pm 0.38}$	$\mathbf{52.74 \pm 0.20}$	$\mathbf{58.90 \pm 0.64}$	$\mathbf{64.79 \pm 0.06}$

Table 2: Accuracy of ChimeraMix with AutoAugment (AA) and TrivialAugment (TA) on ciFAIR-100. ciFAIR-10 and STL-10 are shown in the supplementary material.

input image and is defined as

$$\mathcal{L}_{\text{rec}} = ||\hat{x}_1 - x_1||^2 + ||\hat{x}_2 - x_2||^2. \quad (2)$$

It is common to improve the visual appearance of generated images by using a perceptual loss that is computed using the features of a VGG [Simonyan and Zisserman, 2015] network. However, as we are in the small data setting, we have to resort to another method to match the appearance of our outputs and the input images. Our loss $\mathcal{L}_{\text{per}}$ uses a laplacian pyramid of image-output pairs and computes the l^1-distance between them [Denton *et al.*, 2015]. Thus, we minimize the following generator loss

$$\mathcal{L}_{\text{G}} = \alpha_{\text{rec}}\mathcal{L}_{\text{rec}} + \alpha_{\text{per}}\mathcal{L}_{\text{per}} + \alpha_{\text{disc}}\mathcal{L}_{\text{G,disc}}, \quad (3)$$

where α_{rec}, α_{per}, and α_{disc} are weightings. The exact parameters can be found in the supplementary material.

Classification. After ChimeraMix has been trained on a given dataset, it is used to augment the dataset and train an arbitrary classifier. The generative model enables the composition of new images by sampling combinations of image pairs and different masks. In this way, we are able to greatly increase the dataset. For training the classifier, each batch is replaced with a probability of 50% with compositions that are generated by ChimeraMix.

3.2 Masks

We evaluate two different ways of generating the mask that guides the mixing of the image features in ChimeraMix. See Fig. 3 for an example.

Grid. The masks can be sampled as a grid. For that, we sample m from a binomial distribution $\mathcal{B}$ and interpolate it to the size of the feature tensor $H' \times W'$. The dimension of m, i.e., the size of the blocks, is a hyperparameter and depends on the size of the structural components in the image. By generating a grid structure, blocks of features are mixed in the generator while maintaining the local structure of the instances.

Segmentation. However, we can improve on this heuristic by taking the image gradients into account. Classic algorithms such as Felzenszwalb segmentation [Felzenszwalb and Huttenlocher, 2004] enable us to mix the features of contiguous regions in the images. The regions are computed

from a graph representation of the image that is based on pixel similarities. For ChimeraMix, any segmentation algorithm might be used, as the segmentation masks can be precomputed and do not cause significantly more computational cost. We choose Felzenszwalb segmentation as it has only few hyperparameters[1]. Using the segmentation masks, we uniformly sample a segmented region from the image x_1, create the corresponding binary mask, and downsample it to the size of the feature e_1. This leads to more expressive samples. We call this variant *ChimeraMix+Seg*.

(a) Original image   (b) Grid   (c) Segmentation
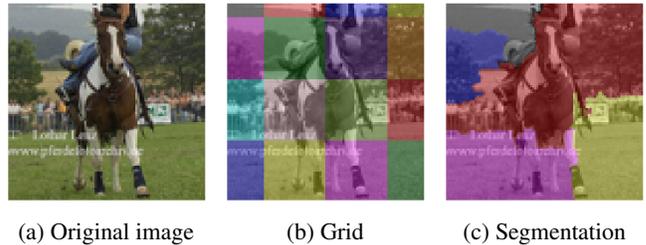
Figure 3: Example image from STL-10 as well as a visualisation of a grid and segmentation mask. For each region, we sample from a binomial distribution. The masks guide the feature mixing in the generator.

4 Experiments

We evaluate the performance of ChimeraMix and current state-of-the-art methods on benchmark datasets and analyze their performance on different dataset sizes. To study the visual quality of the images that ChimeraMix generates, we compare their Fréchet Inception Distance (FID) with that of GLICO while following the methodology of Parmar *et al.* [2021]. Due to the generative nature of ChimeraMix, we can combine it with standard augmentation methods such as AutoAugment and TrivialAugment. Finally, we perform a sensitivity analysis of our approach in the supplementary material and an ablation regarding ChimeraMix's generator.

[1]We chose the initial parameters according to Hénaff *et al.* [2021] and adjusted them to the smaller image sizes.
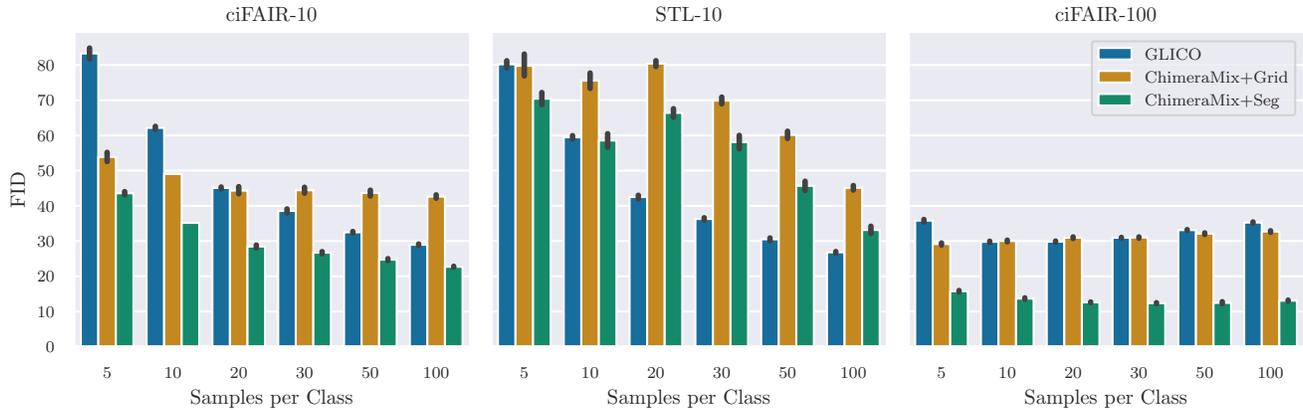
Figure 4: Fréchet Inception Distance($\downarrow$) of GLICO, ChimeraMix+Grid, and ChimeraMix+Seg on ciFAIR-10, STL-10, and ciFAIR-100.

4.1 Datasets

For our experiments, we choose the ciFAIR-10 and ciFAIR-100 [Barz and Denzler, 2020b] datasets. Both contain 50,000 and 10,000 images of size 32×32 in their training and test set and have been extensively used in computer vision research[2]. Lastly, we evaluate our method on STL-10 [Coates *et al.*, 2011] which is a slightly more complex dataset consisting of 5000 training and 8000 test images of size 96×96 from 10 categories. Even though these datasets already are quite small compared to other contemporary datasets such as ImageNet [Russakovsky *et al.*, 2015], we further subsample the labels per class to evaluate the algorithms in the small data regime. For all datasets we sample the instances uniformly from each class.

4.2 Experimental Setup

We use the same architecture for all methods. On ciFAIR-10 and ciFAIR-100 we train a WideResNet-16-8 and on STL-10 we use a ResNet-50 due to the larger image size. We use SGD with momentum and a cosine-annealing learning rate schedule with weight decay. The hyperparameters can be found in the supplementary material. All experiments are repeated five times on different dataset splits.

4.3 Comparison with State-of-the-Art

We compare ChimeraMix on all datasets against several methods such as Cutout [DeVries and Taylor, 2017], Random Erasing [Zhong *et al.*, 2020], Cosine loss [Barz and Denzler, 2020a], MixUp [Zhang *et al.*, 2018], and GLICO [Azuri and Weinshall, 2021]. On ciFAIR-10 and ciFAIR-100 we compare against Parametric Scattering Networks [Gauthier *et al.*, 2021] which showed state-of-the-art in the small data regime on ciFAIR-10[3]. Additionally, we include a comparison of

SuperMix [Dabouei *et al.*, 2021] on ciFAIR-100. For all datasets, a standard classification network is evaluated as a baseline.

The results are shown in Section 1 for different numbers of samples per class. On ciFAIR-10 with 5 samples per class, for example, where only 50 training examples are available, the current best performing method achieves 33.41%. ChimeraMix+Grid and ChimeraMix+Seg reach a test performance of 36.94% and 37.31%, respectively. Overall, the results demonstrate that ChimeraMix+Grid and ChimeraMix+Seg generate image compositions that improve the training and lead to a higher accuracy, especially in the small data regime.

4.4 Comparison with TrivialAugment and AutoAugment

In the next experiment, we investigate whether ChimeraMix can be combined with known methods for automatic data augmentation such as AutoAugment [Cubuk *et al.*, 2019] and TrivialAugment [Müller and Hutter, 2021]. It should be noted, that the policies of AutoAugment are optimized on the entire dataset. The performance of AutoAugment, TrivialAugment, ChimeraMix, and combinations on ciFAIR-100 are shown in Section 2. Evaluations on the other datasets are presented in the supplementary material.

The results show that TrivialAugment achieves a higher accuracy than AutoAugment on ciFAIR-100, and vice versa on ciFAIR-10 and STL-10. ChimeraMix, without any severe data augmentation, is already able to reach a similar performance as AutoAugment. The combination of ChimeraMix and AutoAugment or TrivialAugment significantly increases the performance. On ciFAIR-100 with 5 samples per class, ChimeraMix+Seg achieves 26.48% in combination with TrivialAugment compared to 23.85% without ChimeraMix+Seg.

4.5 Analysis of Generated Samples

We conduct an analysis of the generated samples similar to GLICO [Azuri and Weinshall, 2021] since both are generative approaches. The samples of ChimeraMix should be valid compositions of the given image pairs and thus have similar

---

[2]The ciFAIR-10 and ciFAIR-100 datasets have the same training set as their CIFAR [Krizhevsky, 2009] counterparts, but provide a cleaned test set.

[3]Parametric Scattering Networks uses a modified WideResNet that has more units in the first layers. Since the performance with a standard WideResNet degrades, we evaluate their method with the modified architecture.

Samples per Class		5	10	20	30	50	100
Dataset	Method						
ciFAIR-10	GridMix	29.97 ± 1.17	39.90 ± 1.24	48.60 ± 3.18	54.99 ± 2.49	61.12 ± 1.59	72.41 ± 0.69
	SegMix	32.00 ± 1.22	42.18 ± 1.36	52.56 ± 2.43	57.90 ± 0.49	64.61 ± 0.94	73.96 ± 0.36
	ChimeraMix+Grid	36.94 ± 2.63	45.57 ± 2.11	53.67 ± 2.84	59.66 ± 1.35	65.42 ± 0.83	73.76 ± 0.30
	ChimeraMix+Seg	$\mathbf{37.31 \pm 2.57}$	$\mathbf{47.60 \pm 1.81}$	$\mathbf{56.21 \pm 1.77}$	$\mathbf{60.92 \pm 0.62}$	$\mathbf{67.30 \pm 1.21}$	$\mathbf{74.96 \pm 0.21}$
STL-10	GridMix	28.98 ± 1.49	31.21 ± 1.52	37.08 ± 1.09	42.14 ± 1.52	49.33 ± 0.88	56.92 ± 0.51
	SegMix	29.25 ± 0.40	32.84 ± 0.63	37.80 ± 1.91	43.69 ± 0.84	50.14 ± 0.84	58.60 ± 0.57
	ChimeraMix+Grid	$\mathbf{32.18 \pm 0.90}$	37.01 ± 0.84	43.19 ± 1.03	48.93 ± 1.34	52.81 ± 1.45	60.04 ± 0.27
	ChimeraMix+Seg	31.37 ± 1.72	$\mathbf{37.05 \pm 1.09}$	$\mathbf{44.74 \pm 0.60}$	$\mathbf{49.58 \pm 0.49}$	$\mathbf{55.06 \pm 1.11}$	$\mathbf{60.44 \pm 0.71}$
ciFAIR-100	GridMix	17.98 ± 0.23	27.78 ± 0.46	38.92 ± 0.05	45.16 ± 1.05	52.97 ± 0.30	61.37 ± 0.26
	SegMix	19.36 ± 0.82	29.62 ± 0.22	41.00 ± 0.42	47.50 ± 0.38	54.62 ± 0.15	62.43 ± 0.38
	ChimeraMix+Grid	20.24 ± 0.12	31.62 ± 0.82	41.80 ± 0.52	48.10 ± 0.71	54.67 ± 1.01	62.13 ± 0.27
	ChimeraMix+Seg	$\mathbf{21.09 \pm 0.47}$	$\mathbf{32.72 \pm 0.60}$	$\mathbf{43.23 \pm 0.38}$	$\mathbf{48.83 \pm 0.72}$	$\mathbf{55.79 \pm 0.21}$	$\mathbf{62.96 \pm 0.77}$

Table 3: Analysis of the generator's impact. GridMix and SegMix directly mix the images without the generator of ChimeraMix. The study shows that mixing the feature via the proposed generator (ChimeraMix+Grid and ChimeraMix+Seg) is able to learn the generation of new image compositions and achieves a significantly improved performance.

visual features. Fig. 2 shows qualitative example outputs of ChimeraMix on STL-10. The generator has learned to combine the two images of a cat in sensible ways, e.g., by placing the head of one cat on the body of the other cat. If these features are diverse enough and close to the original full training set, the classifier should be able to achieve a similar validation accuracy. We compare the features of our generated datasets by using an Inception-v3 [Szegedy *et al.*, 2015] network that was pretrained on ImageNet and compute the FID [Heusel *et al.*, 2017] to the original full training set[4]. The FID is defined as

$$\text{FID} = ||\mu - \mu_\omega||_2^2 + \text{tr}\left(\Sigma + \Sigma_\omega - 2\left(\Sigma^{\frac{1}{2}}\Sigma_\omega\Sigma^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) \quad (4)$$

comparing the gaussian approximations of the activations between the two datasets. μ_ω and Σ_ω are the mean and covariance of the full training set.

The results are shown in Fig. 4 and indicate that ChimeraMix is able to produce samples that are close to the full training set. Compared to the GLICO baseline, except for STL-10, we achieve better FID scores. A visualization of the representation of the augmented dataset is shown in the supplementary material.

4.6 Ablation Study

To analyze the effect of our generator-discriminator architecture, we compare ChimeraMix with two versions *GridMix* and *SegMix* that mix the images directly using our two methods to produce masks. Given two images and a mask, we are already able to generate compositions and train a classifier on them. Note, that GridMix is similar to CutMix by Yun *et al.* [2019] except that the patches consist of multiple possibly overlapping rectangles. The results are shown in Section 4.4. It is apparent that the generator of ChimeraMix improves the quality of the mixed images and leads to a significant improvement. On ciFAIR-10 with 10 samples per

class, for example, ChimeraMix+Grid increases the accuracy from 39.90% to 45.57% and ChimeraMix+Seg from 42.18% to 47.60%.

5 Conclusion

In this work, we presented ChimeraMix, a novel method to improve the performance of image classifiers given only a handful of samples per class. Our generative approach mixes the features of two images from the same class using a binary mask and learns to generate new samples that are compositions of the given images. The mask is either sampled from a binomial distribution or generated from the segmented areas in both images. The experiments show that ChimeraMix is able to generate new image compositions that significantly improve the classification. We demonstrate state-of-the-art classification performance on several datasets and investigate the distribution of the compositions that our method generates using the Fréchet Inception Distance. In future work, our method can be extended to mix multiple images. Additionally, the masks can be refined by integrating techniques such as SuperMix after they have been adjusted to the small data setting.

Acknowledgments

References

[Arora *et al.*, 2020] Sanjeev Arora, Simon S. Du, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, and Dingli Yu. Harnessing the power of infinitely wide deep nets on small-data tasks. In *Proc. of ICLR*, 2020.

---

[4]We use the code provided by Parmar *et al.* [2021].

[Assran *et al.*, 2021] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, 2021.

[Awiszus *et al.*, 2020] Maren Awiszus, Frederik Schubert, and Bodo Rosenhahn. Toad-gan: Coherent style level generation from a single example. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 16, 2020.

[Azuri and Weinshall, 2021] I. Azuri and D. Weinshall. Generative latent implicit conditional optimization when learning from small sample. In *25th International Conference on Pattern Recognition, ICPR*, 2021.

[Barz and Denzler, 2020a] Bjorn Barz and Joachim Denzler. Deep Learning on Small Datasets without Pre-Training using Cosine Loss. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, 2020.

[Barz and Denzler, 2020b] Björn Barz and Joachim Denzler. Do we train on test data? Purging CIFAR of near-duplicates. *Journal of Imaging*, 6, 2020.

[Bornschein *et al.*, 2020] Jörg Bornschein, Francesco Visin, and Simon Osindero. Small data, big decisions: Model selection in the small-data regime. In *Proc. of ICML*, 2020.

[Brigato and Iocchi, 2021] Lorenzo Brigato and Luca Iocchi. A close look at deep learning with small data. *25th International Conference on Pattern Recognition, ICPR*, 2021.

[Coates *et al.*, 2011] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. volume 15 of *Proceedings of Machine Learning Research*. PMLR, 2011.

[Cubuk *et al.*, 2019] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[Dabouei *et al.*, 2021] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M. Nasrabadi. SuperMix: Supervising the mixing data augmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

[Denton *et al.*, 2015] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems 28*, 2015.

[DeVries and Taylor, 2017] Terrance DeVries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. *ArXiv preprint*, abs/1708.04552, 2017.

[Felzenszwalb and Huttenlocher, 2004] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[Gauthier *et al.*, 2021] Shanel Gauthier, Benjamin Thérien, Laurent Alsène-Racicot, Irina Rish, Eugene Belilovsky, Michael Eickenberg, and Guy Wolf. Parametric Scattering Networks. *ArXiv preprint*, abs/2107.09539, 2021.

[Hénaff *et al.*, 2021] Olivier J. Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, 2021.

[Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30*, 2017.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, 2015.

[Kluger *et al.*, 2018] Florian Kluger, Christoph Reinders, Kevin Raetz, Philipp Schelske, Bastian Wandt, Hanno Ackermann, and Bodo Rosenhahn. Region-based cycle-consistent data augmentation for object detection. In *2018 IEEE International Conference on Big Data*, 2018.

[Kolesnikov *et al.*, 2020] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *Computer Vision – ECCV 2020*. 2020.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 2012.

[Krizhevsky, 2009] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[Müller and Hutter, 2021] Samuel Müller and Frank Hutter. TrivialAugment: Tuning-free yet state-of-the-art data augmentation. *ArXiv*, abs/2103.10158, 2021.

[Neyshabur *et al.*, 2020] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems 33*, 2020.

[Parmar *et al.*, 2021] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On Buggy Resizing Libraries and Surprising Subtleties in FID Calculation. *ArXiv preprint*, abs/2104.11222, 2021.

[Reinders *et al.*, 2018] Christoph Reinders, Hanno Ackermann, Michael Ying Yang, and Bodo Rosenhahn. Object recognition from very few training examples for enhancing bicycle maps. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[Reinders *et al.*, 2019] Christoph Reinders, Hanno Ackermann, Michael Ying Yang, and Bodo Rosenhahn. Learning convolutional neural networks for object detection with very little training data. *Multimodal Scene Understanding*, 2019.

[Renard *et al.*, 2020] Félix Renard, Soulaimane Guedria, Noel De Palma, and Nicolas Vuillerme. Variability and reproducibility in deep learning for medical image segmentation. *Scientific Reports*, 10(1), 2020.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 2015.

[Schubert *et al.*, 2021] Frederik Schubert, Maren Awiszus, and Bodo Rosenhahn. Toad-gan: a flexible framework for few-shot level generation in token-based games. *IEEE Transactions on Games*, 2021.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR*, 2015.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 2014.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2015.

[Yun *et al.*, 2019] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. CutMix: Regularization strategy to train strong classifiers with localizable features. In *IEEE/CVF International Conference on Computer Vision, ICCV*, 2019.

[Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *Proc. of ICLR*, 2018.

[Zhong *et al.*, 2020] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, 2020.