# DIFFUSEMIX: Label-Preserving Data Augmentation with Diffusion Models

Khawar Islam[1] Muhammad Zaigham Zaheer[2] Arif Mahmood[3] Karthik Nandakumar[2]

[1]FloppyDisk.AI [2]Mohamed bin Zayed University of Artificial Intelligence [3]Information Technology University, Punjab

[1]khawarr.islam@gmail.com [2]{zaigham.zaheer, karthik.nandakumar}@mbzuai.ac.ae [3]arif.mahmood@itu.edu.pk

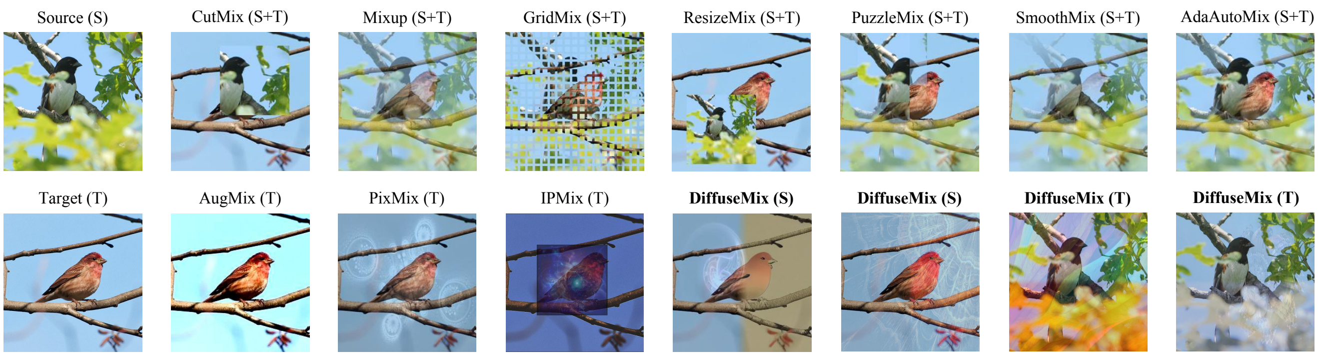



Figure 1. **Top row:** existing mixup methods *interpolate* two different training images [22, 49]. **Bottom row:** label-preserving methods. For each input image, DIFFUSEMIX employs *conditional prompts* to obtain generated images. The input image is then concatenated with a generated image to obtain a hybrid image. Each hybrid image is blended with a random fractal to obtain the final training image.

## Abstract

*Recently, a number of image-mixing-based augmentation techniques have been introduced to improve the generalization of deep neural networks. In these techniques, two or more randomly selected natural images are mixed together to generate an augmented image. Such methods may not only omit important portions of the input images but also introduce label ambiguities by mixing images across labels resulting in misleading supervisory signals. To address these limitations, we propose* DIFFUSEMIX*, a novel data augmentation technique that leverages a diffusion model to reshape training images, supervised by our bespoke conditional prompts. First, concatenation of a partial natural image and its generated counterpart is obtained which helps in avoiding the generation of unrealistic images or label ambiguities. Then, to enhance resilience against adversarial attacks and improve safety measures, a randomly selected structural pattern from a set of fractal images is blended into the concatenated image to form the final augmented image for training. Our empirical results on seven different datasets reveal that* DIFFUSEMIX *achieves superior performance compared to existing state-of-the-art methods on tasks including general classification, fine-grained classification, fine-tuning, data scarcity, and adversarial robustness. Augmented datasets and codes are available here: https://diffusemix.github.io/*

## 1. Introduction

In the era of deep learning, image-mixing-based data augmentation techniques stand out for their simplicity and effectiveness in addressing the generalization of learning models toward testing scenarios [3, 6, 19, 21–23, 23, 35, 41, 47, 51]. These techniques ingeniously mix randomly selected natural images and their respective labels from the training dataset using a number of *mixing* combinations to synthesize new augmented images and labels. Such a process often implies linear interpolation of data, resulting in the generation of novel training images. These approaches have been proven to be effective in improving the performance of deep models [13, 19, 37, 46, 49],.

However, these techniques may face a number of challenges such as the omission of salient image regions (Figure 1) and label ambiguities due to random placements of images [23]. A few researchers have attempted to alleviate these issues by introducing saliency-based mixup strategies in which important regions of one image are pasted onto the less important portions (mainly context) of another image [21, 23, 41]. These methods not only suffer from the costs but also the shortcomings of saliency detection methods. Moreover, as these methods still rely on mixing of two or more images belonging to different classes, the underlying issue of omitting the important context still persists.

Recently, Diffusion Models [11, 12, 32, 38, 39] have emerged as transformative approaches offering image-to-

Table 1. Comparison of different image mixing techniques: most methods utilize natural images as source and target except [42] using hidden state. DiffuseMix uses a *generated* image produced by a diffusion model leveraging *conditional prompts* and a fractal image for augmentation.

| | Input | Mixup [49] | ManifoldMixup [42] | CutMix [46] | SaliencyMix [41] | StyleMix [18] | PuzzleMix [23] | CoMixup [22] | PixMix [17] | GuidMixup [21] | **DiffuseMix** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Components* | Source image | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Target image | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| | Fractal image | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| | Textual Prompts | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| | Interpolation | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| | Concatenation | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| *Tasks* | Adversarial Robustness | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| | General Classification | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Fine Grained | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |
| | Transfer Learning | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| | Data Scarcity | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |

image generation and editing processes. Although the idea of using images generated by diffusion models directly as augmented images for training a classifier has been studied by some researchers [1, 40], this way of data augmentation does not result in significant performance gains. In fact, as reported in [1], the model trained using generated images directly as augmentation may even result in lower performance than the baseline trained without any augmentation. The underlying problem can be attributed to the limited control that these diffusion models offer over generated images. Owing to the sensitivity of diffusion models to conditional prompts, generation of desired complex *scenes*, *layouts*, and *shapes* in an image is a cumbersome task [50]. Thus, poorly constructed prompts pose the risk of producing images that may not be suitable for data augmentation as the generated images may deviate drastically from the actual data distribution. Training on such images may result in overfitting of the learning model on wrong data distribution, consequently resulting in performance degradation. Therefore, careful selection of prompts for data augmentation needs further investigation. Moreover, to mitigate the risk of poorly generated images affecting the overall training, a more efficient way of utilizing the generated images is necessary.

To this end, we propose a novel data augmentation method, DiffuseMix, that leverages the capabilities of a Stable Diffusion model to generate diverse samples based on our tailored conditional prompts. In contrast to Trabucco et al. [40], rather than solely relying on Stable Diffusion for augmentation, we propose an effective approach which utilizes both original and generated images to create hybrid images. This way, visual diversity obtained by the diffusion models is *infused* with the original images while retaining the key semantics. In addition, to increase overall structural diversity, we blend self-similarity-fractals with the hybrid images to create the final training images. This blending has previously been found useful for ML safety measures [17, 20] while in our approach, this added diversity helps in avoiding overfitting on the generated contents resulting in performance improvements. Our experimental results show that DiffuseMix benchmarks better generalization as well as increased adversarial robustness compared to the existing state-of-the-art (SOTA) augmentation methods. Moreover, it offers compatibility with a broad spectrum of datasets and can be incorporated into the training of various existing architectures. Some notable aspects of this research work are as follows:

- We introduce a new data augmentation method driven by a diffusion model, which generates diverse images via our bespoke conditional prompts.
- We propose to *concatenate a portion of the natural image with its generative counterpart* to obtain hybrid images. The combination brings richer visual appearances while preserving key semantics.
- We collect a fractal image dataset and blend into the hybrid images. This improves the overall structural complexity of the augmented images and helps to avoid overfitting on generated images, thus resulting in better generalization.
- Extensive experiments on seven datasets for various tasks including *general classification, fine-grained classification, adversarial robustness, transfer learning, and data scarcity* demonstrate the superior performance of our proposed method compared to the existing SOTA image augmentation techniques.

## 2. Related Work

Data augmentation has become indispensable in enhancing the diversity of training datasets, thereby mitigating the risks of overfitting. Traditional approaches employed strategies—such as horizontal and vertical translations, affine transformations, scaling, and squeezing—when training a model. This not only improves the performance but also improves the generalization of the model on test datasets.

**Diffusion Models for Augmentation:** Recently, several re-

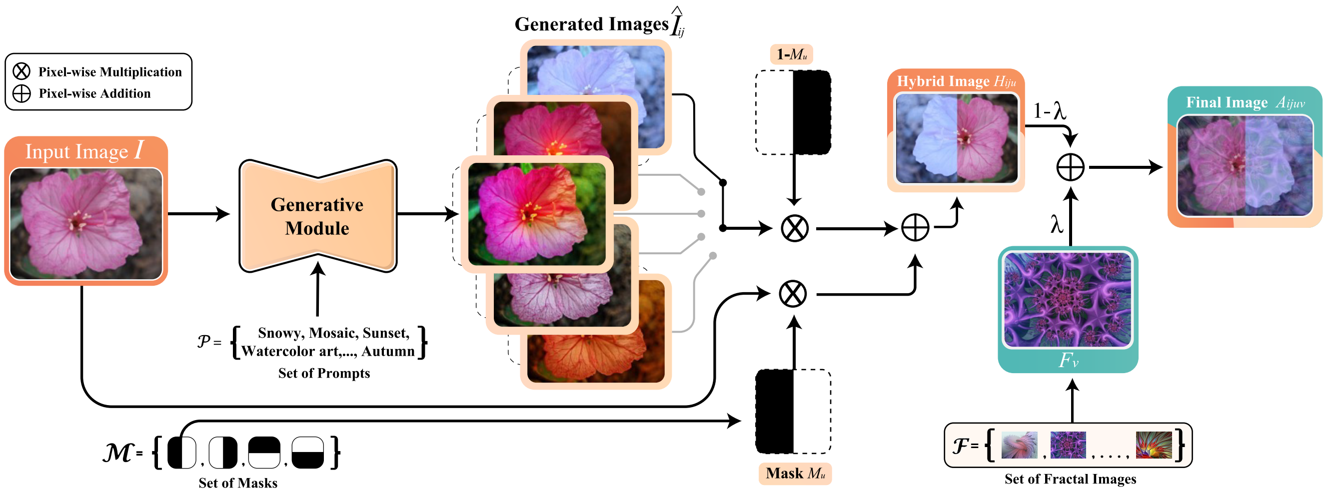



Figure 2. **Architecture of the proposed DiffuseMix approach**. An input image and a randomly selected prompt are input to a diffusion model to obtain a generated image. Input and generated images are concatenated using a binary mask to obtain a hybrid image. A random fractal image is finally blended with this hybrid image to obtain the augmented image.

searchers have explored the possibility of data augmentation with diffusion models. Azizi et al. [1] proposed the utilization of fine-tuned text-to-image diffusion models on ImageNet classification, revealing that augmenting the training set with these synthetic samples may boost classification performance. Similarly, Trabucco et al. [40] investigated diffusion models to create more diverse and semantically varied datasets, aiming to improve outcomes in tasks such as image classification. Li et al. [29] further explored diffusion models-based augmentation for knowledge distillation without real images.

**Image Mixing Augmentation:** Image mixing is a prominent class of augmentation methods for training robust CNN models [4, 30, 34, 45]. Some of these methods include Mixup, CutMix, and AugMix. Mixup [48] generates synthetic images by linearly interpolating pixel values from two randomly selected images. In contrast, CutMix [47] involves pasting a random patch from one image onto another. AugMix [15] employs a stochastic combination of data augmentation operations on an input image. SaliencyMix [41] utilizes saliency maps to concentrate the augmentation on the image's most vital regions, ensuring overall image integrity. Manifold Mixup [42] enhances representation by interpolating network hidden states during training. This entails blending two hidden states with a random weight to produce an interpolated manifold-based hidden state. PuzzleMix [23], an improvement over the traditional mixup, factors in image saliency, and local statistics during image blending. This method segments an image into patches, allocates weights based on saliency and local statistics, and merges patches from different images in accordance with their weights. PixMix [17] have studied mixing of input images with fractal and feature visualization images to improve ML safety measures. A detailed summary of several image-mixing-based methods along with their components and application tasks is provided in Table 1.

**Automated Augmentation:** AutoAugment [7], for instance, employed reinforcement learning to pinpoint optimal data augmentation policies, while RandAugment [9] integrates a suite of random data augmentation operations to improve model generalization. AdaAug[5] is proposed to efficiently learn adaptive augmentation policies in a class-dependent and potentially instance-dependent manner.

In contrast to previous methods, our approach emphasizes the concatenation of original and generated images, using a pre-defined library of conditional prompts. The obtained hybrid images are blended with fractal images to further improve the overall performance.

## 3. DiffuseMix

### 3.1. Background and Overview

Existing image-mixing-based methods may induce label ambiguity by placing one image on top of the other and consequently overlapping either some portions of the object or its context [21, 46]. In contrast, the core idea of DiffuseMix is to concatenate a portion of the original image with its counterpart generated image in such a way that basic image semantics are preserved while providing diverse object details and contexts for better augmentation.

The proposed method as illustrated in Figure 2 comprises of three pivotal steps: ***generation***, ***concatenation***, and ***fractal blending***. Firstly, conditional prompts are used with a diffusion model to obtain a generative counterpart of the input image. Then, a portion of the original image is concatenated with the rest of the portion taken from the generated image forming a hybrid image. This step is to ensure that the training network always has access to the

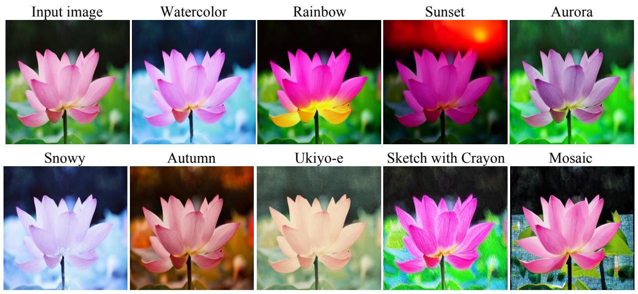



Figure 3. A set of *bespoke conditional prompts* are used to obtain generated images preserving important features and adding rich visual appearance to the input images.

original data along with the generated one. Subsequently, a random fractal image is blended into the hybrid image to obtain the final training image with a diverse structure. Blending fractal images has proven to be effective towards ML safety [16, 17]. In our work, we study the effectiveness of blending fractal images mainly towards improved performance.

### 3.2. Method

The proposed DIFFUSEMIX is an effective data augmentation technique which can be used to enhance the robustness and generalization of the deep learning models. Formally, $I_i \in \mathbb{R}^{h\times w\times c}$ is an image from the training dataset, $\mathcal{D}_{\text{mix}}(\cdot) : \mathbb{R}^{h\times w\times c} \rightarrow \mathbb{R}^{h\times w\times c}$ denotes our data augmentation method. To obtain the final augmented image $A_{ijuv}$, input image $I_i$ goes through proposed generation using prompt $p_j$, concatenation using mask $M_u$, and blending using fractal image $F_v$. The overall augmentation process, as also seen in Algorithm 1, can be represented as $A_{ijuv} = \mathcal{D}_{\text{mix}}(I_i, p_j, M_u, F_v, \lambda)$.

**Generation:** Our generation step $\mathcal{G}(.)$ consists of a *pretrained* diffusion model that takes a prompt $p_j$ from a predefined set of $k$ prompts, $P = \{p_1, p_2, \ldots, p_k\}$ where $j \in [1, k]$ along with the input image $I_i$ and produces an augmented counterpart image $\hat{I}_{ij}$. Image editing process in conventional diffusion models is often open-ended and guided by text prompts to obtain diverse image-to-image or text-to-image translations. In our case, as the goal is to achieve a slightly modified but not too different version of $I_i$, *filter-like* prompts are curated in $\mathcal{P}$ which do not alter the image drastically. Examples of the prompts used in DIFFUSEMIX are shown in Figure 3. The overall generation step can be represented as: $\hat{I}_{ij} = \mathcal{G}(I_i, p_j)$, where $p_j$ is a randomly selected prompt.

**Concatenation:** We concatenate a portion of the original input image $I_i$ with its counterpart generated image $\hat{I}_{ij}$ using a randomly selected mask $M_u$ from the set of masks to create a hybrid image $H_{iju}$:

$$H_{iju} = (\hat{I}_{ij} \odot M_u) + (I_i \odot (\mathbf{1} - M_u)). \quad (1)$$

**Algorithm 1** DIFFUSEMIX

**Require:** $I_i \in \mathcal{D}$ training images dataset, $m$: number of augmented images, $p_j \in \mathcal{P}$ set of prompts, $M_u \in \mathcal{M}$ set of masks, $F_v \in \mathcal{F}$ library of fractal images, $\lambda$: blend ratio

**Ensure:** $\mathcal{D}'$: $m$ Augmented images

$\mathcal{D}' \leftarrow \emptyset$
**for** each image $I_i$ in $\mathcal{D}$ **do**
  **for** $a$ in $\{1 : m\}$ **do**
    Randomly select prompt $p_j$ from $\mathcal{P}$
    Generate image: $\hat{I}_{ij} \leftarrow \mathcal{G}(I_i, p_j)$
    Randomly select mask $M_u$ from $\mathcal{M}$
    Hybrid image: $H_{iju} \leftarrow M_u \odot I_i + (1 - M_u) \odot \hat{I}_{ij}$
    Randomly select $F_v$ from $\mathcal{F}$
    Blended image: $A_{ijuv} \leftarrow (1 - \lambda)H_{iju} + \lambda F_v$
    Add $A_{ijuv}$ to $\mathcal{D}'$
  **end for**
**end for**
**return** $D'$

The mask $M_u$ consists of zeros and ones only and $\odot$ is a pixel-wise multiplication operator. The set of masks contains four kinds of masks including horizontal, vertical and flipped versions. Such masking ensures the availability of the semantics of the input image to the learning network while reaping the benefits of the generated images.

**Fractal Blending:** A fractal image dataset* $\mathcal{F}$ is collected and used for inducing structural variations in the hybrid images. A randomly selected fractal image $F_v \in \mathcal{F}$ is blended to the hybrid image $H_{iju}$ with a blending factor $\lambda$ as:

$$A_{ijuv} = \lambda F_v + (1 - \lambda)H_{iju}, \quad (2)$$

where $\lambda$ is the blending factor. This results in the final augmented image $A_{ijuv}$ used to train or fine-tune a deep learning model. The overall augmentation process of DIFFUSEMIX can be represented as:

$$A_{ijuv} = (1 - \lambda)(I_i \odot M_u + \hat{I}_{ij} \odot (\mathbf{1} - M_u)) + \lambda F_v, \quad (3)$$

## 4. Experiments and Results

In this section, we present the experimental details, datasets used to evaluate our approach, and analyses of the results.

**Datasets.** To provide comparisons with existing studies on image augmentation [5, 8, 9, 15, 19, 21–23, 31, 41, 42, 46, 49], we evaluate our approach on several *general image classification* and *fine-grained image classification* datasets. In the general image classification category, we employ three datasets including ImageNet [10], CIFAR100 [25] and

*Examples of fractal images are provided in Appendix 8.

Tiny-ImageNet-200 [26]. In fine-grained image classification category, we employ four datasets including Oxford-102 Flower [36], Stanford Cars [24], Aircraft [33], and Caltech-UCSD Birds-200-2011 (CUB) [43]. These datasets offer a diverse array of scenarios where images contain a wide range of objects such as plants and animals in various scenes, textures, transportation modes, human actions, satellite imagery, and general objects.

**Implementation Details.** We utilize InstructPix2Pix [2] diffusion model to generate images with the help of our introduced textual library. For the generation of Mask $M$ in Eq. 1, a template image is divided into two equal parts, either horizontally or vertically. Randomly, one half is turned on and the second half is turned off. In all experiments †, $\lambda = 0.20$ is used for blending the fractal image in Eq. (2) & (3). Analysis on $\lambda$ values is provided in Appendix 1.

**Textual Prompt Selection.** In order to ensure that only appropriate prompts are applied, a bespoke textual library of *filter-like* global visual effects is predefined: *'autumn', 'snowy', 'sunset', 'watercolor art', 'rainbow', 'aurora', 'mosaic', 'ukiyo-e', and 'a sketch with crayon'*. These prompts are selected because of their generic nature and applicability to a wide variety of images. Secondly, these do not alter the image structure significantly while producing a global visual effect in the image. Each prompt in the textual library is appended with a template 'A transformed version of image into *prompt*' to form a particular input to the diffusion model. Examples of images generated through these prompts are shown in Figure 3. Additional visual examples and discussions on appropriate prompt selection are provided in Appendix 3.

**Visualizing Intermediate Steps.** Figure 4 depicts images obtained in each step of DIFFUSEMIX. It can be observed that DIFFUSEMIX yields a broader spectrum of augmented images derived from the training set. These images contain full object with no portions omitted and provide suitable variations for training.

# 5. Performance Evaluation

## 5.1. General Classification

General Classification (GC) can be a notable way to quantify the effectiveness of an image augmentation method. A higher GC accuracy would mean that an augmentation method successfully provided plausible data variations to improve the learning process. We evaluate our approach for the GC task on three challenging datasets including Tiny-ImageNet-200, CIFAR-100, and ImageNet. Similar to the existing SOTA methods, PreActResNet18 network is employed for the experiments on Tiny-ImageNet-200 and CIFAR-100 datasets [19, 21, 22, 41], while ResNet50 is employed for ImageNet dataset [15, 23, 46].

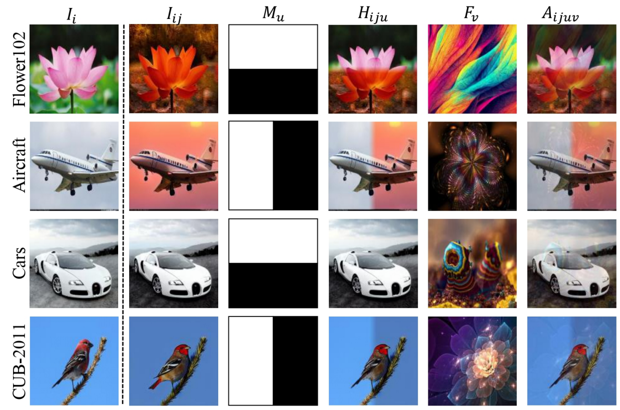



Figure 4. Example images from different stages of DIFFUSEMIX: input image ($I_i$), generated image ($\hat{I}_{ij}$), mask ($M_u$), hybrid image ($H_{iju}$), fractal image ($F_v$), and final augmented image ($A_{ijuv}$).

In Table 2 Top-1 and Top-5 accuracy on Tiny-ImageNet and CIFAR-100 datasets is compared with the existing SOTA methods. The proposed DIFFUSEMIX technique demonstrates better performance gains compared to the existing image augmentation approaches. On Tiny-ImageNet dataset, compared to Vanilla model, our DIFFUSEMIX results in notable Top-1 and Top-5 accuracy gains of 8.54% and 10.01%. Moreover, compared to the second best performer, Guided-AP [21], Top-1 and Top-5 accuracy gains of our approach are 1.14% and 1.17%. Similar trends are observed on CIFAR100 dataset, where DIFFUSEMIX demonstrates Top-1 and Top-5 accuracy gains of 6.17% and 4.39% over vanilla and 1.3% and 0.53% over Guided-AP.

We also evaluate DIFFUSEMIX on large-scale ImageNet dataset offering more challenging scenarios where existing SOTA methods [15, 23, 46] only report Top-1 accuracy. Compared to the second best performer PuzzleMix [23], our approach demonstrates a performance gain of 1.13% (Table 3). Compared to Vanilla implementation, our approach demonstrates a performance gain of 2.95%. The GC results on these challenging and diverse benchmark datasets highlight the effectiveness of DIFFUSEMIX in enabling better learning. It also suggests its capability to combat overfitting and achieve better generalization.

## 5.2. Adversarial Robustness

Following existing SOTA methods [15, 18, 21–23, 41, 42, 46], we evaluate the robustness of our approach against adversarial attacks and input perturbations. In these experiments, fast adversarial training [44] is adapted to create adversarially perturbed input images. The goal of these experiments is to evaluate whether an augmentation approach can demonstrate better resilience against adversarial attacks. FGSM [44] error rates are computed to evaluate the performance against adversarial attacks.

†Following AugMix [15], we employ JS-divergence during training.

Table 2. Top-1 and Top-5 accuracy on ***general classification task*** of PreactResNet-18 trained from scratch for 300 epochs following the results of Kang and Kim [21]. Extended table can be seen in Appendix 7 Table 12.

| Method | Tiny-ImageNet-200 | | CIFAR-100 | |
|---|---|---|---|---|
| | Top-1 (%) | Top-5 (%) | Top-1 (%) | Top-5 (%) |
| Vanilla(CVPR'16) [14] | 57.23 | 73.65 | 76.33 | 91.02 |
| SaliencyMix(ICLR'21) [41] | 56.54 | 76.14 | 79.75 | 94.71 |
| Guided-SR(AAAI'23) [21] | 55.97 | 74.68 | 80.60 | 94.00 |
| PuzzleMix(ICML'20) [23] | 63.48 | 75.52 | 80.38 | 94.15 |
| Co-Mixup(ICLR'21) [22] | 64.15 | - | 80.15 | - |
| Guided-AP(AAAI'23) [21] | 64.63 | 82.49 | 81.20 | 94.88 |
| **DIFFUSEMIX** | **65.77** | **83.66** | **82.50** | **95.41** |

Table 3. Top-1 / Top-5 performance on ImageNet-1K dataset benchmark when trained on ResNet-50 for 100 epochs for ***general classification task***. An extended version of this table is provided in Appendix 7 Table 13.

| Method | Top-1 (%) | Top-5 (%) |
|---|---|---|
| Vanilla(CVPR'16) [14] | 75.97 | 92.66 |
| PixMix(CVPR'22) [17] | 77.40 | - |
| PuzzleMix(ICML'20) [23] | 77.51 | 93.76 |
| GuidedMixup(AAAI'23) [21] | 77.53 | 93.86 |
| Co-Mixup (ICLR'21) [22] | 77.63 | 93.84 |
| YOCO(ICML'22) [13] | 77.88 | - |
| **DIFFUSEMIX** | **78.64** | **95.32** |

Table 4. FGSM error rates on CIFAR-100 and Tiny-ImageNet-200 datasets for PreactResNet-18, following [23].

| Method | FGSM Error Rates (%) | |
|---|---|---|
| | CIFAR-100 | Tiny-ImageNet-200 |
| Vanilla(CVPR'16) [14] | 23.67 | 42.77 |
| Mixup (ICLR'18) [49] | 23.16 | 43.41 |
| Manifold (ICML'19) [42] | 20.98 | 41.99 |
| CutMix (ICCV'19) [46] | 23.20 | 43.33 |
| AugMix (ICLR'20) [15] | 43.33 | - |
| PuzzleMix(ICML'20) [23] | 19.62 | 36.52 |
| **DIFFUSEMIX** | **17.38** | **34.53** |

As shown in Table 4, DIFFUSEMIX demonstrates an error rate of 17.38% on CIFAR-100, which is lower than all compared methods. PuzzleMix is the second best performer obtaining 19.62% error rate. Similarly on Tiny ImageNet-200, DIFFUSEMIX outperforms SOTA by a notable margin obtaining 34.53% error rate while PuzzleMix remained the second best performer with 36.52% error rate. These results demonstrate that even under adversarial perturbations, DIFFUSEMIX remains resilient surpassing the performance of existing SOTA approaches.

### 5.3. Fine-Grained Visual Classification

Compared to the general classification, the task of Fine-Grained Visual Classification (FGVC) is notably challenging since it is difficult for a learning model to identify subtle differences between two different objects belonging to the same general class. A robust image augmentation technique should preserve these subtle albeit critical details for the learning model to successfully train on the fine-grained classification task. To evaluate DIFFUSEMIX on this task, we conduct experiments on three datasets including CUB [36], Stanford Cars [24], and Aircraft [33] utilizing ResNet-50 [14] network.

As shown in Table 5, DIFFUSEMIX significantly enhances the generalization capability of ResNet-50, yielding *superior* performances on all benchmark datasets. Specifically, DIFFUSEMIX outperforms widely-acknowledged methods such as Mixup and CutMix, with notable margins. On CUB dataset, DIFFUSEMIX achieved an accuracy of 79.37%, which is a significant leap from the 65.50% Vanilla accuracy. It also outperformed recent methods including Guided-AP 77.08%, and PuzzleMix 72.83%. Similarly, on the Aircraft dataset, DIFFUSEMIX obtained an accuracy of 85.76% surpassing the second best, GuidedMixup 84.32%. The Stanford Cars dataset further validates the remarkable performance of DIFFUSEMIX registering 91.26% accuracy. On the same dataset, other methods such as SnapMix and PuzzleMix achieved 90.10% and 89.68% respectively. The consistent performance gains on various challenging datasets iterate the effectiveness of DIFFUSEMIX in retaining critical salient information necessary to perform fine-grained classification.

Table 5. Top-1 (%) performance comparison on ***fine-grained task*** of ResNet-50. Extended comparisons are provided in Appendix 7 Table 14.

| Method | Birds | Aircraft | Cars |
|---|---|---|---|
| Vanilla(CVPR'16) [14] | 65.50 | 80.29 | 85.52 |
| RA(NIPS'20) [9] | - | 82.30 | 87.79 |
| AdaAug(ICLR'22) [5] | - | 82.50 | 88.49 |
| Mixup(ICLR'18) [49] | 71.33 | 82.38 | 88.14 |
| CutMix(ICCV'19) [46] | 72.58 | 82.45 | 89.22 |
| SnapMix(AAAI'21) [19] | 75.53 | 82.96 | 90.10 |
| PuzzleMix(ICML'20) [23] | 74.85 | 82.66 | 89.68 |
| Co-Mixup(ICLR'21) [22] | 72.83 | 83.57 | 89.53 |
| Guided-AP(AAAI'23) [21] | 77.08 | 84.32 | 90.27 |
| **DIFFUSEMIX** | **79.37** | **85.76** | **91.26** |

### 5.4. Transfer Learning

Transfer learning or fine-tuning is a widely used way of customizing large architectures with limited computational resources as well as for quick experiments. Most image-mixing-based augmentation methods [15, 15, 19, 21, 21–23, 41, 42, 46, 49] have not reported performance in this important scenario. Nevertheless, we evaluate the perfor-

Table 6. Top-1 (%) accuracy on ***data scarcity*** task of ResNet-18 on Flower102 dataset where only 10 random images per class are used. Extended comparisons are provided in Appendix 7 Table 15.

| Method | Valid | Test |
|---|---|---|
| Vanilla (CVPR'16) [14] | 64.48 | 59.14 |
| SnapMix (AAAI'21) [19] | 65.71 | 59.79 |
| PuzzleMix (ICML'20) [23] | 71.56 | 66.71 |
| Co-Mixup (ICLR'20) [22] | 68.17 | 63.20 |
| GuidedMixup (AAAI'23) [21] | 74.74 | 70.44 |
| **DIFFUSEMIX** | **77.14** | **74.12** |

mance of our approach on fine-tuning the baseline model using three different datasets including Flower102, Aircraft, and Stanford Cars on ImageNet-pretrained ResNet-50 provided by PyTorch and report results in Table 8.

For the Flower102 dataset, DIFFUSEMIX achieved an accuracy of 98.02%, which is higher than the second best method, AdaAug 97.19%. A similar trend is observable in the Aircraft 85.65% and Cars 93.17% datasets. We also observe that the accuracy obtained by DIFFUSEMIX with fine-tuning (Table 8) is comparable to the performance when training is done from scratch (Table 5). Since fine-tuning consumes significantly less computational resources compared to training from scratch, this experiment elaborates the practical significance of DIFFUSEMIX.

### 5.5. Data Scarcity

Data scarcity is one of the major issues when training deep neural networks. If the training examples per class are limited, deep networks may not learn meaningful patterns by leading to overfitting and loss of generalization. Augmentation methods are often used to overcome these challenges by generating more data for training. Under this setting, we compare the accuracy of ResNet-18 [14] trained using only 10 images per class of the original Flower102 dataset. As shown in Table 6, our method consistently outperforms other mixup methods in situations where data is limited yielding accuracies of 77.14% Top-1 and 74.12% Top-5. Furthermore, our approach is designed to enhance the diversification of the training dataset. By leveraging our bespoke conditional prompts, DIFFUSEMIX artificially expands and enriches the overall data landscape, enabling a more robust learning of neural networks.

### 5.6. Analysis and Discussions

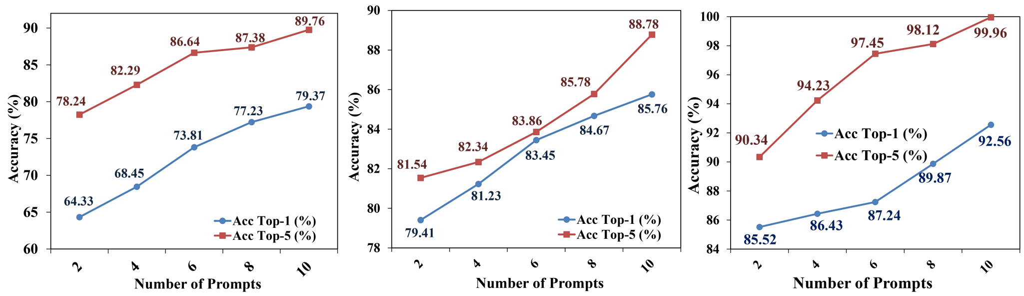
In this section, we provide a detailed ablation study and further analysis of various design choices in DIFFUSEMIX.

**Ablation Studies:** To evaluate the importance of each component we conduct an ablation study by removing each component and report the observed performance on ResNet-50 using Stanford Cars and Flowers-102 datasets in Table 7.

When all of the components of DIFFUSEMIX are removed, the baseline (vanilla) using only original images $I_i$ obtains Top-1 and Top-5 accuracies of 85.52% & 90.34% on Cars dataset and 78.83% & 94.38% on Flowers dataset. Next, we add fractal blending ($F_v$) to the input images resulting in slight performance gains on both datasets. Further, we remove both concatenation ($H_{iju}$) and fractal blending ($F_v$) by conducting experiments using generated images ($\hat{I}_i j$) directly as augmented images to train the network. This setting brings the method closer to the approaches proposed in [1, 40] which utilize the images generated using diffusion models directly as augmented images. In this experiment, the accuracies obtained on Cars dataset are 87.63% Top-1 and 90.23% Top-5. Similarly, we observe the accuracies of 77.38% and 93.15% on Flowers102 dataset. The results are consistent with the findings in [1] that direct use of generated images may not yield significant performance gains over the vanilla method. Further, we carry out experiments by using generated images with fractal blending to train the model. On the Cars dataset, we observe slight performance gains over the vanilla model with accuracies of 89.42% Top-1 and 91.57% Top-5. However, in Flowers dataset, we observe a slight performance degradation. This demonstrates that to unleash the maximum benefits of fractal blending, it should be accompanied by more diversity in the training dataset. Next, we just remove the fractal blending from DIFFUSEMIXwhile incorporating concatenation between generated ($\hat{I}_i j$) and original ($I_i$) images to create hybrid images ($H_{iju}$) used as augmented training examples. This setting increases the accuracies to 90.59% Top-1 & 96.73% Top-5 on Cars dataset and 79.22% Top-1 & 94.38% Top-5 on Flowers dataset. These performance improvements are due to the availability of both generated and original image contents in each augmented image, highlighting the importance of the concatenation step in DIFFUSEMIX. Finally, when all components including fractal blending are used, the best accuracies of 91.26% Top-1 & 99.96% Top-5 on Cars dataset and 80.20% Top-1 & 95.40% Top-5 on Flowers dataset are achieved.

Table 7. Ablation study using Stanford Cars (cars) and Flowers102 (Flow) datasets. Top-1 and Top-5 accuracies are reported with ***different combinations*** of $I_i$: Input image, $\hat{I}_{ij}$: Generated images using prompts $p_j$, $H_{iju}$: Hybrid images using random mask $M_u$, and $F_v$: fractal images used to obtain final blended image $A_{ijuv}$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | $I_i$ | ✓ | ✓ | - | - | - | - |
| | $\hat{I}_{ij}$ | - | - | ✓ | ✓ | - | - |
| | $H_{iju}$ | - | - | - | - | ✓ | ✓ |
| | $F_v$ | - | ✓ | - | ✓ | - | ✓ |
| Cars | Top-1 | 85.52 | 86.73 | 87.63 | 89.42 | 90.59 | **91.26** |
| | Top-5 | 90.34 | 92.38 | 90.23 | 91.57 | 96.73 | **99.96** |
| Flow | Top-1 | 78.73 | 78.34 | 77.38 | 77.81 | 79.22 | **80.20** |
| | Top-5 | 94.38 | 94.91 | 93.15 | 93.24 | 94.38 | **95.40** |




(a) CUB-200-2011 (b) FGVC Aircraft (c) Stanford Cars

Figure 5. Effect of the number of prompts on overall performance. A detailed ablation study showcases the gains in Top-1 (%) and Top-5 (%) accuracy across CUB Birds-200, Aircraft, and Stanford Cars datasets with an increase in the number of prompts in DIFFUSEMIX.

Table 8. Top-1 (%) accuracy of DIFFUSEMIX on ***fine-tuning*** experiments using ImageNet pretrained ResNet-50.

| Method | Flower102 | Aircraft | Cars |
|---|---|---|---|
| Vanilla(CVPR'16) [14] | 94.98 | 81.60 | 88.08 |
| AA(CVPR'19) [8] | 93.88 | 83.39 | 90.82 |
| RA(NIPS'20) [9] | 95.23 | 82.98 | 89.28 |
| Fast AA(NIPS'19) [31] | 96.08 | 82.56 | 89.71 |
| AdaAug(ICLR'22) [5] | 97.19 | 83.97 | 91.18 |
| **DIFFUSEMIX** | **98.02** | **85.65** | **93.17** |

Table 9. Ablation on the ***effects of masking*** in DIFFUSEMIXon Flower102 dataset. All variants yield notably superior results compared to vanilla on ResNet-50. However, best results are achieved when all four vertical and horizontal masks are used.

| Mask | Top-1 (%) | Top-5 (%) |
|---|---|---|
| Vanilla(CVPR'16) [14] | 89.74 | 94.38 |
| Ver Mask ( ) | 94.02 | 98.42 |
| Hor + Ver Masks ( , ) | 94.27 | 99.03 |
| Hor + Ver + Flipping ( , , , ) | 95.37 | 99.39 |

Overall, consistent gains achieved with each added component signifies the design choices in DIFFUSEMIX towards an effective image augmentation technique.

**Number of Prompts Vs. Performance:** The variety of augmented images is determined by the number of random prompts which is an important factor in DIFFUSEMIX. A higher number of prompts corresponds to more diverse training samples. Figure 5 shows the effect of increasing the number of prompts on the performance using three different datasets including Birds, Aircraft and Cars. Increasing the number of prompts consistently yields performance gains on all three datasets using both Top-1 and Top-5 accuracy. However, the peak performances are achieved when all ten prompts proposed in our approach are used for training. In almost all datasets, the increasing accuracy trends can be seen even at 10 prompts which shows that the addition of more prompts may further improve the performance. However, it will also incur more computational costs.

**Increasing Masks Vs Performance:** We conduct experiments using Oxford Flower102 dataset to study the impact of various masks on the overall performance of DIFFUSEMIXand report the results in Table 9. Using even only one kind of mask, vertical in this example, our approach achieves significantly higher accuracies than the vanilla baseline. When both vertical and horizontal masks are used, the accuracies improve further. However, the best accuracies are achieved when both horizontal and vertical masks are used along with random flipping between the positions of input and generated images adding more diversity to the training data. It is also possible to use the masking techniques from previous approaches, such as [27, 46], in DIFFUSMIX. We provide additional analysis on this in Appendix 4.

## 6. Conclusion

In this paper, we introduced DIFFUSEMIX, a data augmentation technique based on diffusion models to increase diversity in the data while preserving the original semantics of the input image. It involves *generation, concatenation, and fractal blending* steps to create the final augmented image. On multiple tasks such as general classification, fine-grained classification, data scarcity, fine-tuning, and adversarial robustness involving several benchmark datasets including *ImageNet-1k, Tiny-ImageNet-200, CIFAR-100, Oxford Flower102, Caltech Birds, Stanford-Cars, and FGVC Aircraft*, DIFFUSEMIXdemonstrates consistent performance gains and outperforms existing SOTA image augmentation methods.

**Limitations:** DIFFUSEMIXhas two notable limitations: (1) Image generation relies heavily on text prompts and a wrong textual input may lead to unrealistic results. We address this issue by proposing a set of *filter-like* prompts generally applicable to a wide range of natural images. (2) DIFFUSEMIXrequires additional overheads for generating images (more on this in Appendix 2). This is a small price to pay for a guaranteed better convergence of large-scale classification models and can be mitigated by generating and storing augmented images once.

**Acknowledgements:** We are thankful to Hamza Saleem for the fruitful insights. Arif Mahmood is funded by the Information Technology University of Punjab, Pakistan.

## References


[1] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[3] Jie-Neng Chen, Shuyang Sun, Ju He, Philip HS Torr, Alan Yuille, and Song Bai. Transmix: Attend to mix for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2022.

[4] Yuan-Chih Chen and Chun-Shien Lu. Rankmix: Data augmentation for weakly supervised learning of classifying whole slide images with diverse sizes and imbalanced categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23936–23945, 2023.

[5] Tsz-Him Cheung and Dit-Yan Yeung. Adaaug: Learning class-and instance-adaptive data augmentation policies. In *International Conference on Learning Representations*, 2021.

[6] Hyeong Kyu Choi, Joonmyung Choi, and Hyunwoo J Kim. Tokenmixup: Efficient attention-guided token-level data augmentation for transformers. *Advances in Neural Information Processing Systems*, 35:14224–14235, 2022.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[8] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[12] Chengbin Du, Yanxi Li, Zhongwei Qiu, and Chang Xu. Stable diffusion is untable. *arXiv preprint*, 2023.

[13] Junlin Han, Pengfei Fang, Weihao Li, Jie Hong, Mohammad Ali Armin, , Ian Reid, Lars Petersson, and Hongdong Li. You only cut once: Boosting data augmentation with a single cut. In *International Conference on Machine Learning (ICML)*, 2022.

[14] K. He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*.

[16] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*, 2021.

[17] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16783–16792, 2022.

[18] Minui Hong, Jinwoo Choi, and Gunhee Kim. Stylemix: Separating content and style for enhanced data augmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14862–14870, 2021.

[19] Shaoli Huang, Xinchao Wang, and Dacheng Tao. Snapmix: Semantically proportional mixing for augmenting fine-grained data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1628–1636, 2021.

[20] Zhenglin Huang, Xiaoan Bao, Na Zhang, Qingqi Zhang, Xiao Tu, Biao Wu, and Xi Yang. Ipmix: Label-preserving data augmentation method for training robust classifiers. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Minsoo Kang and Suhyun Kim. Guidedmixup: an efficient mixup strategy guided by saliency maps. In *AAAI*, pages 1096–1104, 2023.

[22] JangHyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *International Conference on Learning Representations*, 2020.

[23] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.

[24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *CVPRw*, pages 554–561, 2013.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[26] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[27] Jin-Ha Lee, Muhammad Zaigham Zaheer, Marcella Astrid, and Seung-Ik 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*, pages 756–757, 2020.

[28] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Dada: Differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*, 2020.

[29] Zheng Li, Yuxuan Li, Penghai Zhao, Renjie Song, Xiang Li, and Jian Yang. Is synthetic data from diffusion models ready for knowledge distillation? *arXiv preprint arXiv:2305.12954*, 2023.

[30] Wen Liang, Youzhi Liang, and Jianguo Jia. Miamix: Enhancing image classification through a multi-stage augmented mixied sample data augmentation method. *arXiv preprint arXiv:2308.02804*, 2023.

[31] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32, 2019.

[32] Xue-Jing Luo, Shuo Wang, Zongwei Wu, Christos Sakaridis, Yun Cheng, Deng-Ping Fan, and Luc Van Gool. Camdiff: Camouflage image augmentation via diffusion model. *arXiv preprint arXiv:2304.05469*, 2023.

[33] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[34] Thomas Mensink and Pascal Mettes. Infinite class mixup. *arXiv preprint arXiv:2305.10293*, 2023.

[35] Ning Miao, Tom Rainforth, Emile Mathieu, Yann Dubois, Yee Whye Teh, Adam Foster, and Hyunjik Kim. Instance-specific augmentation: Capturing local invariances. 2022.

[36] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[37] Huafeng Qin, Xin Jin, Yun Jiang, Mounim A El-Yacoubi, and Xinbo Gao. Adversarial automixup. *arXiv preprint arXiv:2312.11954*, 2023.

[38] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[39] Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14453–14463, 2023.

[40] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[41] AFM Uddin, Mst Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. *arXiv preprint arXiv:2006.01791*, 2020.

[42] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019.

[43] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[44] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *ICLR*, 2020.

[45] Lingyu Yan, Yu Ye, Chunzhi Wang, and Yun Sun. Locmix: local saliency-based data augmentation for image classification. *Signal, Image and Video Processing*, pages 1–10, 2023.

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *CoRR*, abs/1905.04899, 2019.

[47] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[48] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

[49] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[50] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[51] Qihao Zhao, Yangyu Huang, Wei Hu, Fan Zhang, and Jun Liu. Mixpro: Data augmentation with maskmix and progressive attention labeling for vision transformer. In *The Eleventh International Conference on Learning Representations*, 2022.

# DIFFUSEMIX: Label-Preserving Data Augmentation with Diffusion Models

Supplementary Material

## Overview

This supplementary document contains additional results and discussions. Summarily, Section 1 provides analysis on varying values of $\lambda$ which defines the ratio of blending between a fractal image and a hybrid image in DIFFUSEMIX. Section 2 provides comparisons of augmentation overhead between DIFFUSEMIXand existing image augmentation strategies with respect to their generalization performances. Section 3 discusses examples of poorly constructed prompts and their effects on image generation. Section 4 provides experimental results of using different masking strategies of the state-of-the-art methods with DIFFUSEMIX. Section 5 provides the convergence analysis of DIFFUSEMIX. Section 6 provides more visualizations of the augmented training images obtained using DIFFUSEMIX. Section 7 provides a complete list of general and fined-grained results. Section 8 provides some visual examples of the collected fractal image dataset.

## 1. Fractal Blending Ratio

We experiment and observe the effect of varying fractal blending ratio $\lambda$ in DIFFUSEMIX and report Top-1 accuracy (%) results on Flower102 dataset in Table 10. The values of $\lambda$ are varied from 0.1 to 0.5. A higher value of $\lambda$ indicates a stronger ratio of fractal image blending.

The baseline, ResNet50 without any augmentation, yields a top-1 accuracy of 78.73%. Compared to this, DIFFUSEMIX yields consistent performance gains with all values of $\lambda$. The best performance of DIFFUSEMIX is observed at $\lambda = 0.2$, where the top-1 accuracy peaks at 81.30%. However, generally, the performance remains better with a reasonable value of $\lambda$. It starts dropping when the value of $\lambda$ becomes too high. This suggests that higher fractal blending ratios may introduce too much complexity or noise into the original data, which adversely affects the model's performance.

## 2. Augmentation Overhead

We compare the computational overhead of several existing SOTA image augmentation methods and DIFFUSEMIX with respect to the performance gains. Following Kang *et al.* [21], we define the augmentation overhead $\mathcal{A}_{\mathcal{O}}$ as:

$$\mathcal{A}_{\mathcal{O}} = \frac{\mathcal{T}_{aug} - \mathcal{T}_{van}}{\mathcal{T}_{van}} \times 100(\%),$$

where $\mathcal{T}$ is the total image generation and training time, and $\mathcal{T}_{van}$ is the training time of the baseline network without any augmentation. Although image generation process in DIFFUSEMIX can be expedited by using parallel-processing, We do not utilize it to provide a fair comparison. It can be seen in Figure 6 that DIFFUSEMIX provides a good tradeoff between performance and augmentation overhead by outperforming all existing approaches in terms of accuracy while providing significantly lower augmentation overhead compared to Co-Mixup and SaliencyMix approaches. Moreover, DIFFUSEMIX can also be optimized further by saving the generated images offline once before carrying out any number of subsequent trainings. This may make it significantly faster to perform several experiments on a training model, particularly for optimization and research purposes.

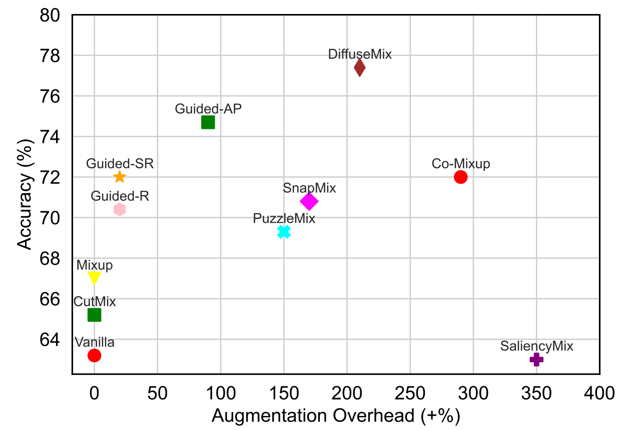


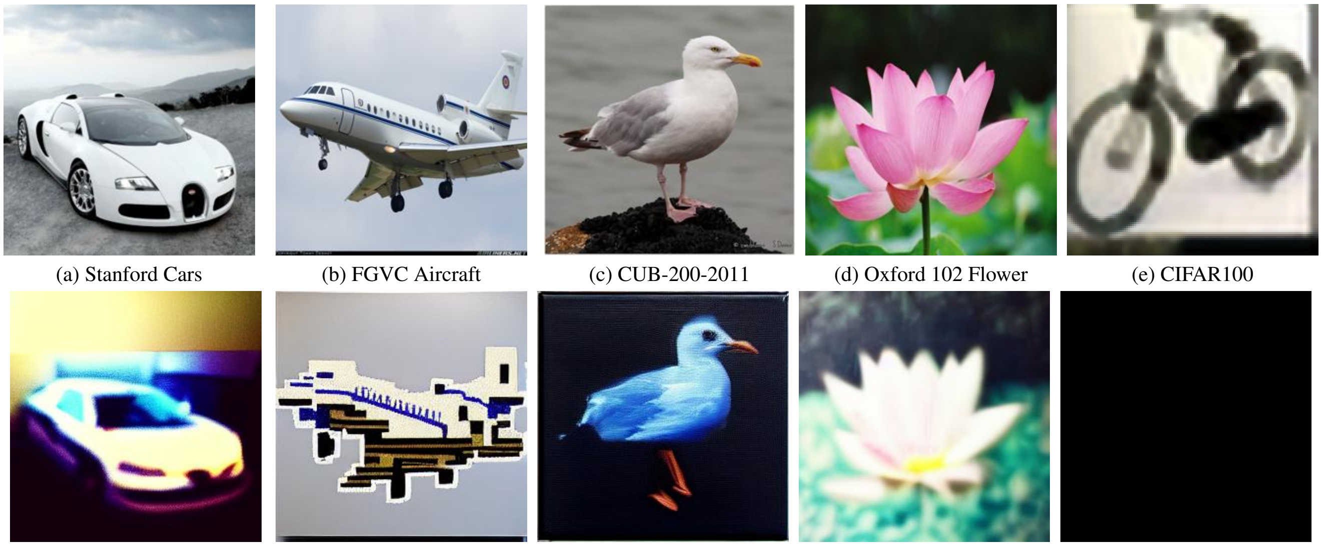

Figure 6. Augmentation overhead (+%) - accuracy (%) plot on CUB-200-2011 dataset with batch size 32.

Table 10. Impact of varying fractal blending ratio in DIFFUSEMIX. Top-1 accuracy is reported using ResNet-50 on Flower102 dataset.

| Methods | Top-1 (%) |
|---|---|
| ResNet50$_{\text{(CVPR'16)}}$ [14] | 78.73 |
| + *DiffuseMix* ($\lambda = 0.1$) | 79.81 |
| + *DiffuseMix* ($\lambda = 0.2$) | **81.30** |
| + *DiffuseMix* ($\lambda = 0.3$) | 80.97 |
| + *DiffuseMix* ($\lambda = 0.4$) | 79.16 |
| + *DiffuseMix* ($\lambda = 0.5$) | 78.97 |

## 3. Prompt Selection

Diffusion Models rely heavily on prompts [12]. Therefore, the intuition behind designing our *bespoke conditional*



(a) Stanford Cars (b) FGVC Aircraft (c) CUB-200-2011 (d) Oxford 102 Flower (e) CIFAR100

***Sunset photo that looks like it's taken with 1990s camera*** ***A painting that is too small for its body*** ***A cityscape that is too dark to see anything*** ***Photograph that is washed it looks like it is made of paper*** ***A distorted, warped painting of a landscape***

Figure 7. **First row:** original training image samples from different datasets *such as* Oxford-102 Flower [36], Stanford Cars [24], and Aircraft [33], CUB-200-2011, and CIFAR100. **Second row:** Corresponding generated images show that the usage of descriptive prompts (blue text) results in poor images not feasible for training. When generating images on the CIFAR100 dataset, several additional challenges may occur due to the small size of the images. For example, the image in the last column taken from CIFAR-100 with its corresponding prompt results in a ***black*** image containing no visible output.

*prompts* is to introduce the type of prompts that may edit the image in a way that preserves structural information and can easily be applied to a range of diverse datasets. To this end, as described in the manuscript, we propose to use *filter-like* prompts such as snowy, sunset, rainbow, etc. and demonstrate their effectiveness in training robust classifiers.

Conversely, in this section, we discuss *bad* prompts that may not be a good fit for the image generation step of DIFFUSEMIX. Some examples of such prompts are shown in Figure 7. More descriptive and overly complicated prompts generate images that may be too different from the original distribution. The resultant images contain unrealistic foregrounds and backgrounds, rendering these useless for the training of a classifier. This reiterates the importance of our proposed *filter-like bespoke conditional prompts* that do not induce unwanted changes to the training images.

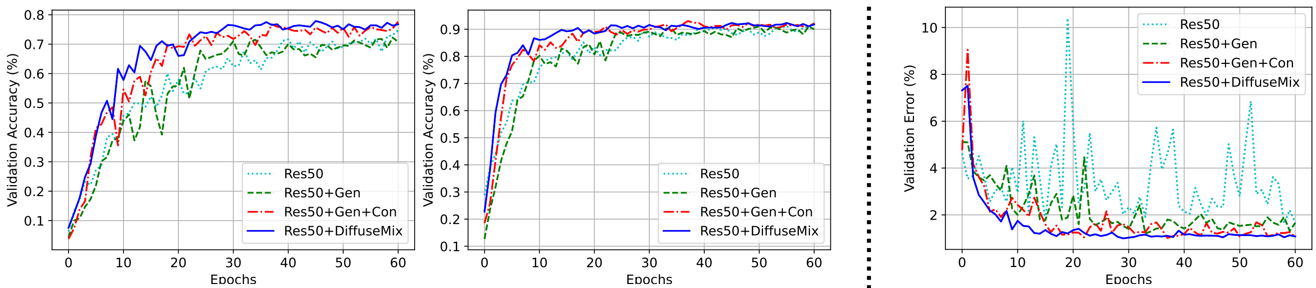
## 4. DIFFUSEMIX with SOTA Methods

In a series of experiments, we combine DIFFUSEMIX with existing image augmentation approaches [46, 49] to see if any performance gain is observed. Particularly, We replace our masking approach with the masking used in the existing methods while retaining the rest of the pipeline of DIFFUSEMIX same.

For CutMix + DIFFUSEMIX, we replace the concatenation step of DIFFUSEMIX with the random cropping of CutMix. To this end, we randomly crop a patch from the generated image and paste it onto the original image whereas the other stages remain the same. For Mixup + DIFFUSEMIX, we replace concatenation with the pixel blending of original and generated images as proposed in [49] while the rest of the steps remain intact. The results are summarized in Table 11. Using CutMix [46] or Mixup [49] methods yields improvements over baseline ResNet50 training. However, when our proposed approach is added to the existing methods, further performance gains are observed. Top performance is finally observed with our DIFFUSEMIX, which demonstrates the importance of forming hybrid images by concatenating original and generated images.

Table 11. Combining DIFFUSEMIX with SOTA image augmentation methods by replacing the image concatenation technique of DIFFUSEMIX with the masking techniques proposed in [46] & [49]. While DIFFUSEMIX provides consistent gains in these settings, the best performance of 81.30% is achieved when our originally proposed method is used.

| Method | Top-1 (%) |
|---|---|
| ResNet50$_{\text{(CVPR'16)}}$ [14] | 78.73 |
| + CutMix [46] | 79.22 |
| + CutMix [46] + DIFFUSEMIX | 79.58 |
| + Mixup [49] | 79.34 |
| + Mixup [49] + DIFFUSEMIX | 80.20 |
| + DIFFUSEMIX | **81.30** |




(a) Top-1 Accuracy (%) (b) Top-5 Accuracy (%) (c) Validation Error (%)

Figure 8. ***On left side:*** The curves of top-1 and top-5 accuracy show an increasing trend during initial 60 epochs and remain stable towards the end on Flower102 dataset. This same behavior can also be seen in top-5 accuracy. —our enables smoother training and better convergence while avoiding overfitting. ***On right side:*** Similar to the accuracy plots, using DIFFUSEMIX demonstrates a smoother decrease in validation error compared to ResNet50 or other variants. *Best viewed in color.*

## 5. DIFFUSEMIX Convergence

**Analysis on Top-1 and Top-5 Accuracy:** In a series of experiments, we carry out an ablation to observe the top-1 and top-5 accuracies of DIFFUSEMIX and its variants formed by removing the components *(generation, concatenation, and fractal blending)* one by one.

As seen in Figures 8a and 8b, the RES50+DIFFUSEMIX demonstrates generally better performance with convergence at 77.26% accuracy, closely followed by DIFFUSEMIX+GEN+CON at 75.79%, and Res50 at 76.41%. The DIFFUSEMIX+GEN model performs significantly lower yielding 73.96% accuracy. As discussed in the manuscript Section 4, using generated images directly for the training may lead to deteriorated performance, which is re-validated in these experiments. This also shows the importance of each step proposed in DIFFUSEMIX towards robust training more robust classifiers. Overall, similar trends are observed in Top-5 accuracy results (Figure 8b).

**Analysis on validation loss:** As seen in (Figure 8c), it is clearly noticeable that DIFFUSEMIX helps in model convergence and overall smooth decrease in validation loss during training. Res50 baseline shows a good start with lower initial loss. However, its loss starts fluctuating once the training is continued indicating a potential plateau in learning or its limitation in capturing more complex patterns. Compared to all variants, Res50+DIFFUSEMIX benchmarks better convergence.

## 6. DIFFUSEMIX Visualizations

In this section, we provide more visual examples of training images obtained using DIFFUSEMIX. As seen in Figure 9, visualizing examples from Flower102 dataset, DIFFUSEMIX enhances the overall variation of the images while retaining the interpretability of each example. For Caltech-UCSD Birds-200-2011 (Figure 10), compared to original images, the augmented images obtained using DIFFUSEMIX exhibit greater clarity and diverse contexts. Similar visual features can be observed in Figure 11 and Figure 12 showcasing examples from Cars and Aircraft datasets, respectively.

Table 12. Top-1 and Top-5 general classification accuracies comparison using PreActResNet-18. Compared numbers are taken either from the original papers or from [21].

| Method | Tiny-ImageNet Top-1 Acc (%) | Tiny-ImageNet Top-5 Acc (%) | CIFAR-100 Top-1 Acc (%) | CIFAR-100 Top-5 Acc (%) |
|---|---|---|---|---|
| Vanilla [14] | 57.23 | 73.65 | 76.33 | 91.02 |
| Mixup [49] | 56.59 | 73.02 | 76.84 | 92.42 |
| Manifold [42] | 58.01 | 74.12 | 79.02 | 93.37 |
| CutMix [46] | 56.67 | 75.52 | 76.80 | 91.91 |
| AugMix [15] | 55.97 | 74.68 | 75.31 | 91.62 |
| PixMix [17] | - | - | 79.70 | - |
| SaliencyMix [41] | 56.54 | 76.14 | 79.75 | 94.71 |
| Guided-SR [21] | 55.97 | 74.68 | 80.60 | 94.00 |
| PuzzleMix [23] | 63.48 | 75.52 | 80.38 | 94.15 |
| Co-Mixup [22] | 64.15 | - | 80.15 | - |
| Guided-AP [21] | 64.63 | 82.49 | 81.20 | 94.88 |
| **DIFFUSEMIX** | **65.77** | **83.66** | **82.50** | **95.41** |

## 7. Performance Evaluation

Extended versions of the performance tables are provided in this section.

### 7.1. General Classification

In Table 12, Vanilla method serves as our baseline, achieving Top-1 accuracies of 57.23% and 76.33% on Tiny-ImageNet and CIFAR-100 respectively, setting a benchmark for subsequent comparisons. For Mixup, we observe a slight decline in performance on Tiny-ImageNet to 56.59% Top-1 accuracy but a marginal improvement on CIFAR-100, reaching 76.84%. Conversely, Manifold Mixup marks

(a) Poinsettia (b) Barbeton Daisy (c) Gazania (d) Dandelion (e) Magnolia

(f) Poinsettia Autumn (g) Barbeton Daisy Snowy (h) Gazania Crayon Sketch (i) Dandelion Crayon Sketch (j) Magnolia Crayon Sketch

Figure 9. Illustration of original training images and DIFFUSEMIX augmented images from the Oxford Flower102 dataset. **First row:** showcases original, unaltered images of various flowers, including *poinsettia, barbeton daisy, gazania, dandelion,* and *Magnolia* classes. **Second row:** illustrates the transformative effects of the DIFFUSEMIX augmentation method. The effects of our custom-tailored prompts-based generation are visible on the generated portion of each image. Overall, DIFFUSEMIX results in a diverse array of images with sufficient structural complexity and diversity to train robust classifiers.

notable performance gains, especially on CIFAR-100 with a Top-1 accuracy of 79.02%. CutMix slightly improves over the baseline on Tiny-ImageNet, whereas AugMix shows a decrement, particularly on CIFAR-100 with a 75.31% Top-1 accuracy.

PixMix introduces variations in the source image instead of mixing two input images. Compared to baseline, PixMix excels on CIFAR-100 with a 79.70% Top-1 accuracy. SaliencyMix, which uses saliency to mix different portions of images, also shows promising results. Particularly on CIFAR-100, it achieves a Top-1 accuracy of 79.75%. The Guided-SR method performs slightly lower compared to AugMix on Tiny-ImageNet but stands out on CIFAR-100 with 80.60% Top-1 accuracy, indicating its effectiveness. PuzzleMix and Co-Mixup introduce more complex ways to augment data, with PuzzleMix reaching a notable 63.48% Top-1 accuracy on Tiny-ImageNet. Co-Mixup tops these methods on Tiny-ImageNet with 64.15% Top-1 accuracy but does not maintain this lead on CIFAR-100. Guided-AP pushes the performance boundaries further by achieving superior accuracies among its predecessors, e.g., 81.20% Top-1 accuracy on CIFAR-100.

DIFFUSEMIX, our proposed method, which surpasses all prior techniques by securing the highest accuracies: 65.77% Top-1 on Tiny-ImageNet and 82.50% Top-1 on CIFAR-100. Our approach not only surpasses the conventional mixup strategies but also sets a new standard in enhancing the generalization of deep learning models. The performance of DIFFUSEMIX stays consistent across the compared datasets, underlining its superior capability and efficiency.

Table 13. Top-1 and Top-5 accuracies comparison on ImageNet using ResNet-50. Compared numbers are taken either from the original papers or from [21].

| Method | Top-1 Acc. | Top-5 Acc. |
|---|---|---|
| Vanilla (CVPR'16) [14] | 75.97 | 92.66 |
| AugMix (ICLR'20) [15] | 76.75 | 93.30 |
| Manifold (ICML'19) [42] | 76.85 | 93.50 |
| Mixup (ICLR'18) [49] | 77.03 | 93.52 |
| CutMix (ICCV'21) [46] | 77.08 | 93.45 |
| Guided-SR (AAAI'23) [23] | 77.20 | 93.66 |
| PixMix (CVPR'22) [17] | 77.40 | - |
| PuzzleMix (ICML'20) [23] | 77.51 | 93.76 |
| GuidedMixup (AAAI'23) [21] | 77.53 | 93.86 |
| Co-Mixup (ICLR'21) [22] | 77.63 | 93.84 |
| YOCO (ICML'22) [13] | 77.88 | - |
| Azizi et al. (arXiV'23) [1] | 78.17 | - |
| **DIFFUSEMIX** | **78.64** | **95.32** |

In Table 13, we provide a comparison of various methods in terms of Top-1 and Top-5 accuracies on ImageNet, specifically when training ResNet-50 as per the training configuration of in Kang and Kim [21]. It starts with the baseline Vanilla ResNet model, showing accuracies of 75.97% for Top-1 and 92.66% for top-5. Various techniques, including Azizi et al., AugMix, Manifold, Mixup, CutMix, Guided-SR, PixMix, PuzzleMix, GuidedMixup,

Table 14. Top-1 accuracy comparison on fine-grained visual classification task while training from scratch on ResNet-50.

| | Methods | Top-1 Accuracy (%) | | |
|---|---|---|---|---|
| | | CUB | Aircraft | Cars |
| | Vanilla [14] | 65.50 | 80.29 | 85.52 |
| *automated* | Auto Aug [8] | - | 82.28 | 88.04 |
| | Fast AA [31] | - | 82.20 | 87.19 |
| | DADA [28] | - | 81.16 | 87.14 |
| | RA [9] | - | 82.30 | 87.79 |
| | AdaAug [5] | - | 82.50 | 88.49 |
| *mixup family* | Mixup [49] | 71.33 | 82.38 | 88.14 |
| | CutMix [46] | 72.58 | 82.45 | 89.22 |
| | SaliencyMix [41] | 66.66 | 83.14 | 89.04 |
| | Guided-SR [21] | 74.08 | 83.51 | 89.23 |
| | SnapMix [19] | 75.53 | 82.96 | 90.10 |
| | PuzzleMix [23] | 74.85 | 82.66 | 89.68 |
| | Co-Mixup [22] | 72.83 | 83.57 | 89.53 |
| | GuidedMixup [21] | 77.08 | 84.32 | 90.27 |
| | **DIFFUSEMIX** | **79.37** | **85.76** | **91.26** |

Co-Mixup, YOCO, and another entry from Azizi et al., display a range of improvements, with Top-1 accuracies spanning from 69.24% to 78.17% and top-5 accuracies (when provided) ranging up to 93.86%. The most notable performance is observed in the DIFFUSEMIX method, which outperforms the others by achieving the highest accuracies at 78.64% for top-1 and 95.32% for top-5.

### 7.2. Fine-Grained Visual Classification

Table 14 presents a comparison of Top-1 accuracy of various methods on a fine-grained visual classification task, using ResNet-50. The methods are categorized into two main groups: automated methods and the mixup family, and are evaluated across three datasets: CUB, Aircraft, and Cars.

In the automated data augmentation, the Vanilla method achieves 65.50%, 80.29%, and 85.52% accuracy on CUB, Aircraft, and Cars respectively. Other automated methods like Auto Aug, Fast AA, DADA, RA, and AdaAug show varied performance, with AdaAug topping this category with accuracies of 82.50% for Aircraft and 88.49% for Cars. The mixup family methods show a notable performance improvement, particularly GuidedMixup demonstrating the accuracies of 77.08%, 84.32%, and 90.27% on the three datasets respectively. Nevertheless, our DIFFUSEMIX stands out by outperforming all compared methods significantly, achieving the highest accuracies of 79.37% for CUB, 85.76% for Aircraft, and 91.26% for Cars.

This indicates that while both categories of methods enhance performances, mixup family methods demonstrate superior capability in handling fine-grained visual classification tasks. DIFFUSEMIX, in particular, showcases exceptional improvements, suggesting its effectiveness in extracting nuanced features from the images.

Table 15. Top-1 accuracy on data scarcity experiment using Flower102 dataset where only 10 random images per class are used. Experiments are performed with ResNet-18 network.

| Methods | Valid (%) | Test (%) |
|---|---|---|
| Vanilla$_{(CVPR'16)}$[14] | 64.48 | 59.14 |
| Mixup$_{(ICLR'18)}$ [49] | 70.55 | 66.81 |
| CutMix$_{(ICCV'19)}$ [46] | 62.68 | 58.51 |
| SaliencyMix$_{(ICLR'21)}$ [41] | 63.23 | 57.45 |
| Guided-SR$_{(AAAI'21)}$ [21] | 72.84 | 69.31 |
| SnapMix$_{(AAAI'21)}$ [19] | 65.71 | 59.79 |
| PuzzleMix$_{(ICML'20)}$ [23] | 71.56 | 66.71 |
| Co-Mixup$_{(ICLR'21)}$ [22] | 68.17 | 63.20 |
| GuidedMixup$_{(AAAI'23)}$ [21] | 74.74 | 70.44 |
| **DIFFUSEMIX** | **77.14** | **74.12** |

### 7.3. Data Scarcity

Table 15 presents the Vanilla method as a baseline with 64.48% accuracy on the validation set and 59.14% on the test set. SOTA techniques like Mixup and PuzzleMix show improved accuracies, with Mixup achieving 70.55% on validation and 66.81% on the test set, and PuzzleMix reaching 71.56% and 66.71%, respectively.

Notably, the Guided-SR and GuidedMixup methods significantly outperform other approaches, with GuidedMixup achieving the highest accuracies of 74.74% on validation and 70.44% on the test set. Our DIFFUSEMIX, which surpasses all compared methods, demonstrates remarkable accuracies of 77.14% on the validation and 74.12% on the test set, showcasing its superior ability to generalize well from significantly limited data. This evidence suggests that data augmentation and mixing techniques, especially DIFFUSEMIX, are highly beneficial in enhancing model performance under stringent data constraints.

### 7.4. Self-Supervised Learning

Table 16 showcases the Top-1 accuracy of self-supervised learning methods, specifically comparing the performance of MoCo v2 and SimSiam.

Initially, MoCo v2 exhibits accuracies of 80.31%, 40.82%, and 51.36% on Flower102, StanfordCars, and Aircraft datasets. After applying DIFFUSEMIX augmentation, it performs better by demonstrating accuracies of

Table 16. Top-1 (%) accuracy of self-supervised learning methods. Adding DIFFUSEMIX yields better performance.

| Method | Flower102 | Stanford Cars | Aircraft |
|---|---|---|---|
| MoCo v2 | 80.31 | 40.82 | 51.36 |
| + DIFFUSEMIX | **82.15** | **41.73** | **53.28** |
| SimSiam | 86.93 | 48.34 | 40.37 |
| + DIFFUSEMIX | **89.24** | **49.17** | **42.63** |

| | Mixup | CutMix | AugMix | Outlier | PixMix | DIFFUSEMIX |
|---|---|---|---|---|---|---|
| Corruptions | 48.0 | 51.5 | 35.4 | 51.5 | 30.5 | **28.5** |
| Consistency | 9.5 | 12.0 | 6.5 | 11.3 | 5.7 | **5.1** |
| Adversaries | 97.4 | 97.0 | 95.6 | 97.2 | 92.9 | **90.2** |
| Calibration | 13.0 | 29.3 | 18.8 | 15.2 | 8.1 | **7.7** |
| Anomaly Det. | 71.7 | 74.4 | 84.9 | **90.3** | 89.3 | 88.3 |

Table 17. On CIFAR-100, DIFFUSEMIX outperforms SOTA on 4 of the 5 distinct safety metrics. Lower is better except for anomaly detection. (SOTA method results are taken from PixMix [17]).

82.15%, 41.73%, and 53.28%. SimSiam starts with accuracies of 86.93%, 48.34%, and 40.37%. Adding DIFFUSEMIX as an augmentation method improves the performance to 89.24%, 49.17%, and 42.63%. This clearly illustrates that integrating DIFFUSEMIX significantly boosts the performance, demonstrating its effectiveness in enhancing self-supervised learning models. The systematic gains across different datasets and on multiple methods highlight the robustness of our approach and its potential to improve the accuracies of different machine learning models.

### 7.5. Safety Measures

Table 17 showcases a comparative analysis of several data augmentation methods on the CIFAR-100 dataset, focusing on their performance across five different safety metrics. The methods evaluated include Mixup, CutMix, AugMix, Outlier, PixMix, and DIFFUSEMIX. The results highlight DIFFUSEMIX's superior performance, as it outperforms the state-of-the-art (SOTA) previously established by PixMix in four out of the five categories. DIFFUSEMIX demonstrates better performance in cases of corruptions, consistency, adversaries, and calibration. In the case of Anomaly detection task, our approach demonstrates comparable performance.

## 8. Fractal Dataset

We collected a dataset of 100 fractal images containing complex patterns and scales. Blending these images to the training images introduces a level of abstraction and complexity not commonly found in regular training images. Some of the example fractal images are provided in Figure 13. As discussed extensively in the manuscript, fractal blending in DIFFUSEMIX helps the network generalize better by adding *contained* noise or perturbations. The ablation studies reported in our manuscript and supplementary suggest that utilizing fractal blending with the generated images helps stabilizing the training and improves the overall convergence.

### 8.1. Fractal with SOTA Methods

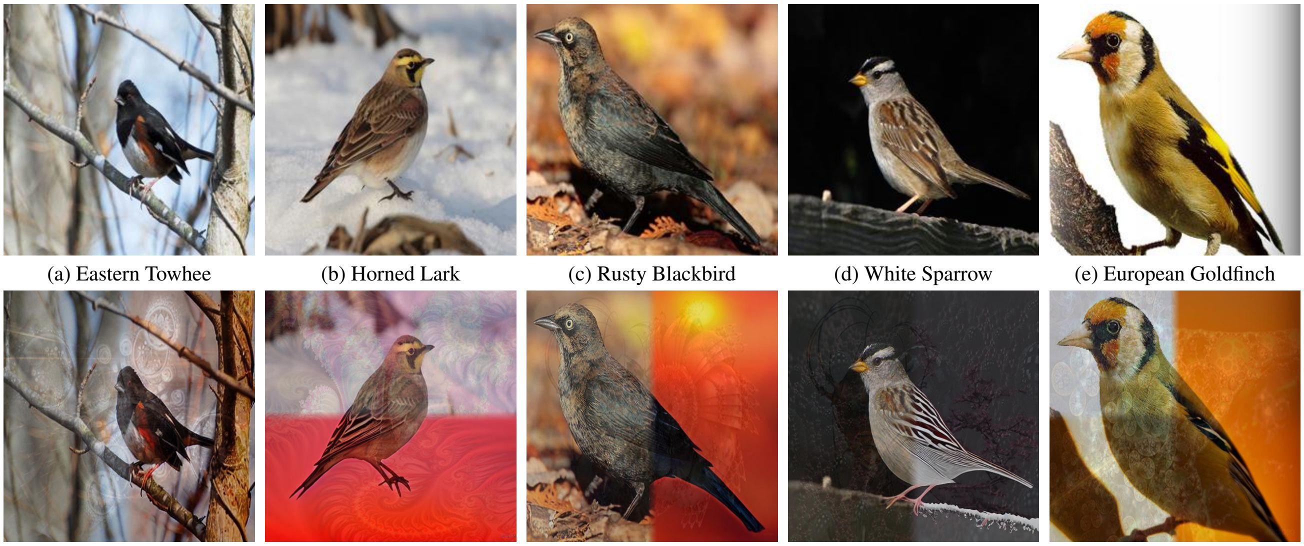
Table 18 presents a performance analysis, particularly focusing on the impact of blending fractals with different augmentation methods using CUB-Birds, Aircraft, Stanford Cars, and Flower102 datasets.

Adding fractal blending to the baseline results in performance improvements on CUB-Birds, Aircraft, and Stanford Cars, but a slight decrease in accuracy on the Flower dataset. The baseline method shows performances of 65.50% on CUB-Birds, 80.29% on Aircraft, 85.52% on Cars and 78.34% on Flower102. The Mixup, CutMix, and PuzzleMix methods, when used without fractal, generally show higher accuracy than the baseline, especially on the Stanford Cars and Aircraft datasets. However, the integration of fractal blending with these methods leads to a significant drop in performance across all datasets, suggesting that fractal blending may not be properly aligned with these particular augmentation techniques.

In contrast, when fractals are blended with the hybrid images (Hybrid $H_{iju}$) in our approach, performance improvements are notably observed in three of the four datasets including Aircraft, Stanford Cars, and Flower102. datasets, this combination leads to improvements in accuracy, indicating a positive synergy between the hybrid images and fractal blending. However, there's a slight decrease in accuracy for the CUB-Birds dataset.

Table 18. Performance comparison (%) of fractal blending with baseline and other augmentation methods, it is more effective when fractals are blended with our hybrid images $H_{iju}$.

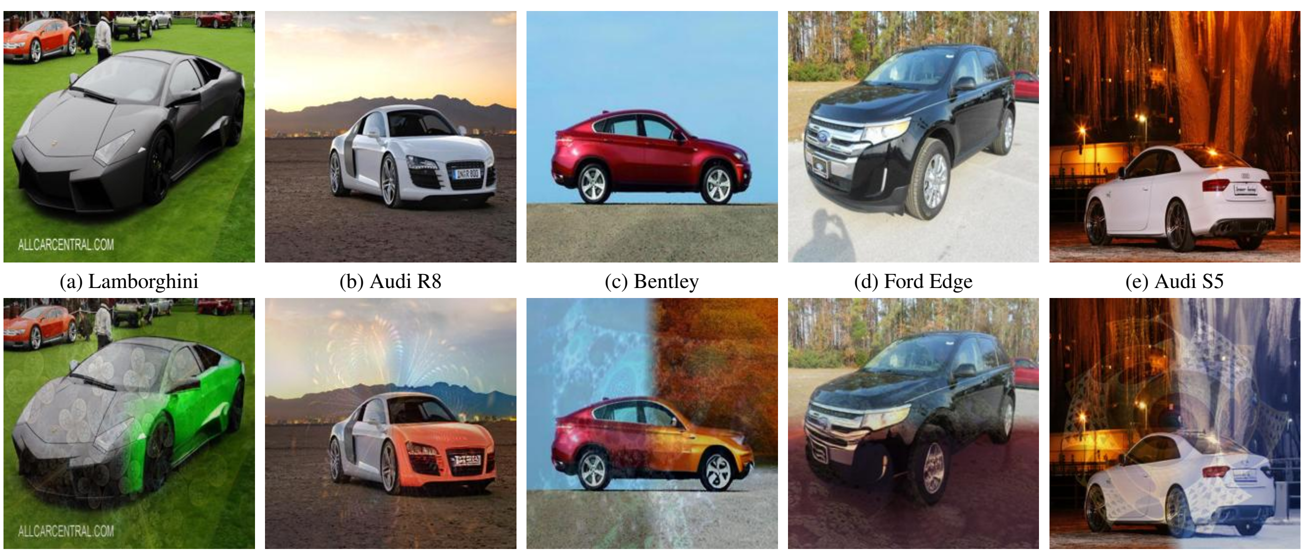
| Method | CUB-Birds | Aircraft | Cars | Flower |
|---|---|---|---|---|
| Baseline | 65.50 | 80.29 | 85.52 | 78.73 |
| + FRACTAL | **66.17** | **81.27** | **86.73** | **78.34** |
| Mixup | 71.33 | 82.38 | 88.14 | 78.12 |
| + FRACTAL | **43.25** | **44.27** | **54.25** | **57.27** |
| CutMix | 72.58 | 82.45 | 89.22 | 74.36 |
| + FRACTAL | **46.74** | **41.47** | **56.37** | **52.28** |
| PuzzleMix | 74.85 | 82.66 | 89.68 | 71.68 |
| + FRACTAL | **51.61** | **53.38** | **61.42** | **63.73** |
| Hybrid ($H_{iju}$) | 80.27 | 85.31 | 90.59 | 79.22 |
| + FRACTAL | **79.37** | **85.76** | **92.56** | **80.20** |



(a) Eastern Towhee (b) Horned Lark (c) Rusty Blackbird (d) White Sparrow (e) European Goldfinch

(f) Eastern Towhee Sunset (g) Horned Lark Sunset (h) Rusty Blackbird Autumn (i) White Sparrow Snowy (j) Europe Goldfinch Snowy

Figure 10. Original and DIFFUSEMIX augmented bird images from the Caltech-UCSD Birds-200-2011 dataset. **Top row:** displays a selection of original, high-resolution bird images, capturing the natural beauty and diversity of species such as the *eastern towhee, horned lark, rusty blackbird, white sparrow,* and *european goldfinch*. **Bottom row:** demonstrates the augmented images obtained using DIFFUSEMIX. The augmented images are visually striking and contextually varied representations of the original subjects.




(a) Lamborghini (b) Audi R8 (c) Bentley (d) Ford Edge (e) Audi S5

(f) Lamborghini Aurora (g) Audi R8 Rainbow (h) Bentley Autumn (i) Ford Edge Sunset (j) Audi S5 Snowy

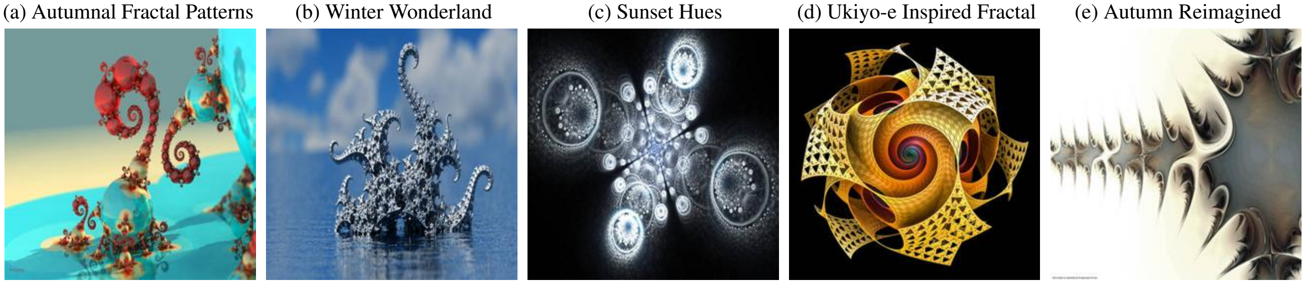
Figure 11. **First row:** showcases original images from the Stanford Cars benchmark dataset, featuring unaltered depictions of various car models including a *lamborghini, audi R8, bentley, ford edge* and *audi S5*. **Second row:** presents the images transformed using our DIFFUSEMIX method. The effects of prompts are visible in the generated portions of the images. For example, *lamborghini* is changed to green when *aurora* prompt is applied, creating a vibrant image. The front side of *audi R8* becomes more color-rich when it is generated with *rainbow* prompt. The ambiance (background context) of *bentley* transforms significantly when *autumn* prompt is used. Similar diverse transformations are observed in other examples. These augmented images demonstrate the capability of DIFFUSEMIX in generating visually enriched augmented images for better generalization.

(a) 737-200 (b) 727-200 (c) 737-700 (d) 777-200 (e) A330-300

(f) 737-200 Sunset (g) 727-200 Autumn (h) 737-700 Snowy (i) 777-200 Ukiyo (j) A330-300 Autumn

Figure 12. Illustration of original and DIFFUSEMIX augmented Aircraft images from the FGVC-Aircraft benchmark dataset. **Top row:** presents original aircraft images, each portraying a distinct airplane including the $737 - 200$, $727 - 200$, $737 - 700$, $777 - 200$, and A$330 - 300$. These images highlight the design resemblance of various aircraft models, serving as a challenging resource for aircraft fined-grained image classification studies. **Bottom row:** showcases the augmented images obtained using DIFFUSEMIX for each corresponding input image. As seen, DIFFUSEMIX reimagined each aircraft with unique prompts such as *sunset, autumn, snowy* and *ukiyo* resulting in a rich visual appearance with diverse contexts. This also illustrates how image augmentation can be used to simulate different environmental and stylistic scenarios, potentially enhancing the robustness and versatility of the dataset for training robust neural networks.

(a) Autumnal Fractal Patterns (b) Winter Wonderland (c) Sunset Hues (d) Ukiyo-e Inspired Fractal (e) Autumn Reimagined



(f) Snowflake Elegance (g) Dusk's Fractal Canvas (h) East Meets West (i) Seasonal Shifts (j) Frozen Fractal Patterns

Figure 13. Some samples taken from our collected fractal dataset. Each subfigure represents a unique fractal image, demonstrating the diversity and complexity inherently present in fractal geometry.