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

CVPR 2024

Introduction

Image-mixing-based data augmentation 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.

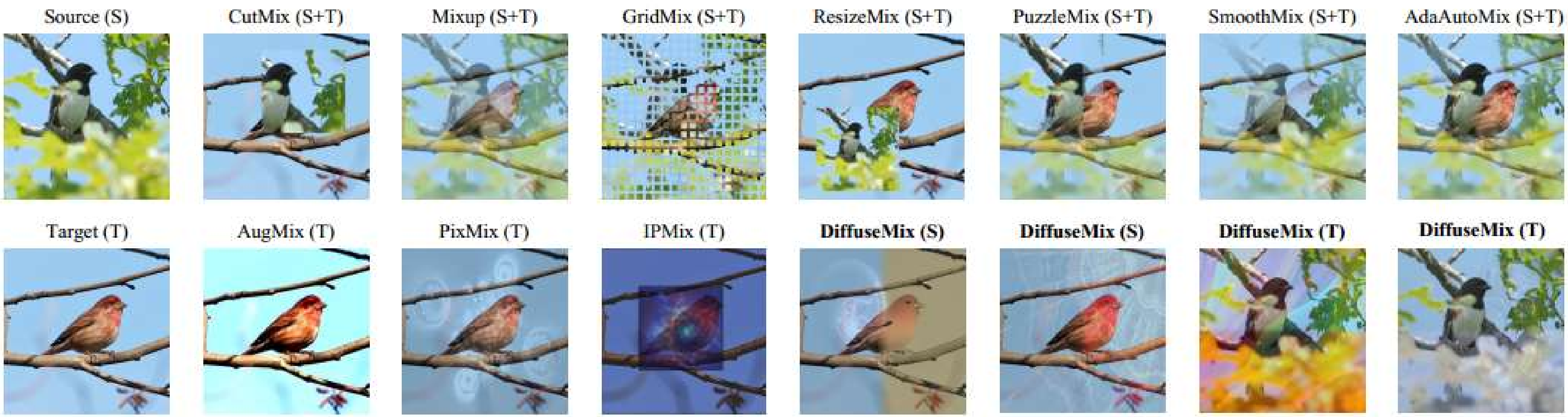


Figure 1. **Top row:** existing mixup methods *interpolate* two different training images [22, 49]. **Bottom row:** label-preserving methods. For each input image, DIFFUSEMIX employs *conditional prompts* to obtain generated images. The input image is then concatenated with a generated image to obtain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

Introduction

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	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓

Method

Overall framework of DIFFUSEMIX

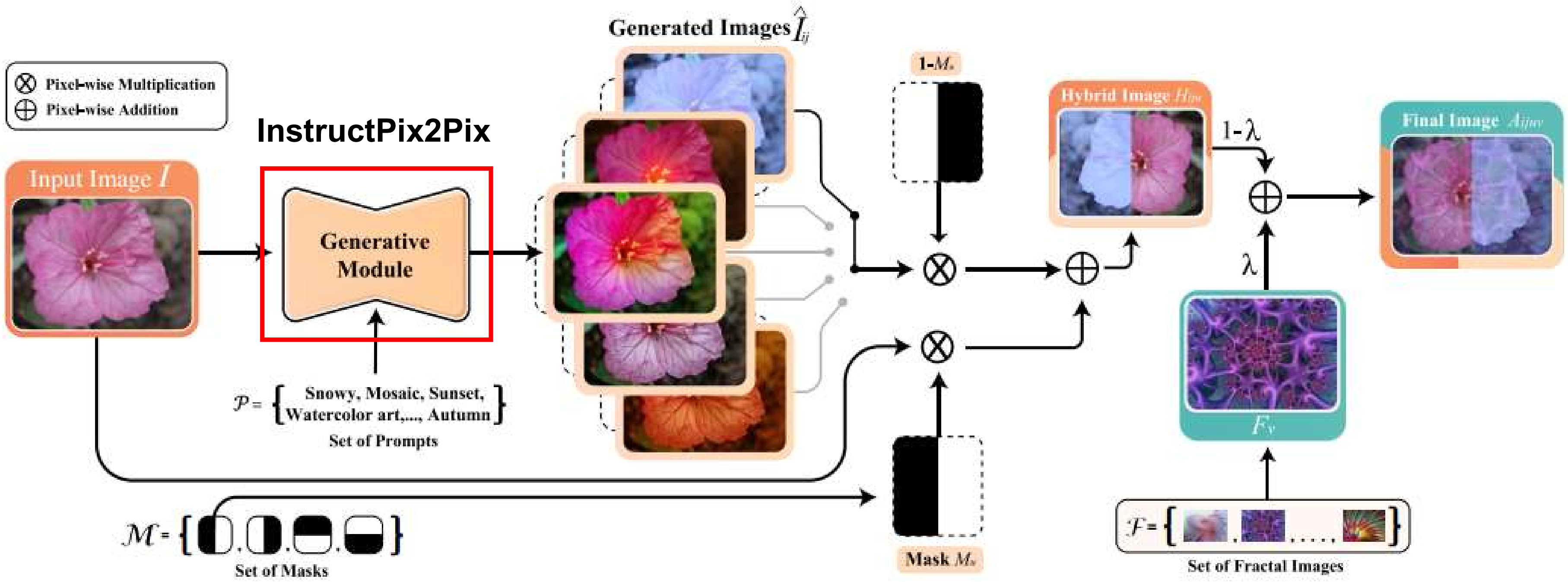


Figure 2. **Architecture of the proposed DIFFUSEMIX approach**. An input image and a randomly selected prompt are input to a diffusion model to obtain a generated image. Input and generated images are concatenated using a binary mask to obtain a hybrid image. A random fractal image is finally blended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

Method

Fractal images is the images generated through fractal geometry that exhibits complex self-similarity and infinite detail.
The generation of fractal images usually relies on simple iterative algorithms.

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

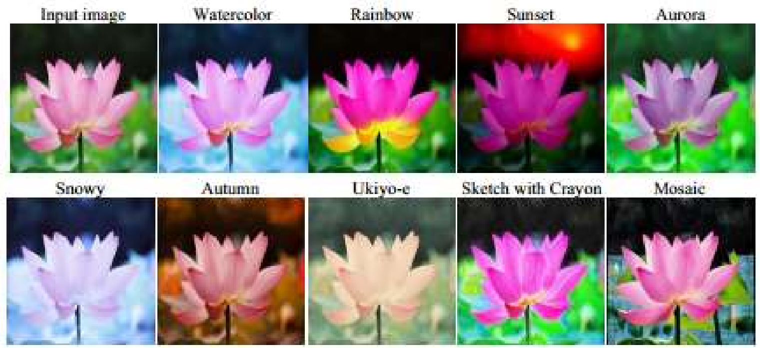
(f) Snowflake Elegance (g) Dusk's Fractal Canvas (h) East Meets West (i) Seasonal Shifts (j) Frozen Fractal Patterns

Method

南京航空航天大學
Nanjing University of Aeronautics and Astronautics

Generation

The generation step consists of a pretrained diffusion model that takes a prompt from a predefined set of k prompts, along with the input image I_i and produces an augmented counterpart image $\hat{I}_{ij}$.



'autumn', 'snowy', 'sunset', 'watercolor art', 'rainbow', 'aurora', 'mosaic', 'ukiyo-e', 'a sketch with crayon'

Concatenation

Concatenate a portion of the original input image with its counterpart generated image by using a randomly selected mask:

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

Fractal Blending

A randomly selected fractal image is blended to the hybrid image with a blending factor λ as:

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

The overall augmentation process of **DIFFUSEMIX** can be represented as:

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

Method

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, λ: blend ratio

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

```
D' ← ∅
for each image I_i in D do
    for a in {1 : m} do
        Randomly select prompt p_j from P
        Generate image: Î_ij ← G(I_i, p_j)
        Randomly select mask M_u from M
        Hybrid image: H_iju ← M_u ⊙ I_i + (1 − M_u) ⊙ Î_ij
        Randomly select F_v from F
        Blended image: A_ijuv ← (1 − λ) H_iju + λ F_v
        Add A_ijuv to D'
    end for
end for
return D'
```

- $\mathcal{D}' \leftarrow \emptyset$
- **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'

Experiments

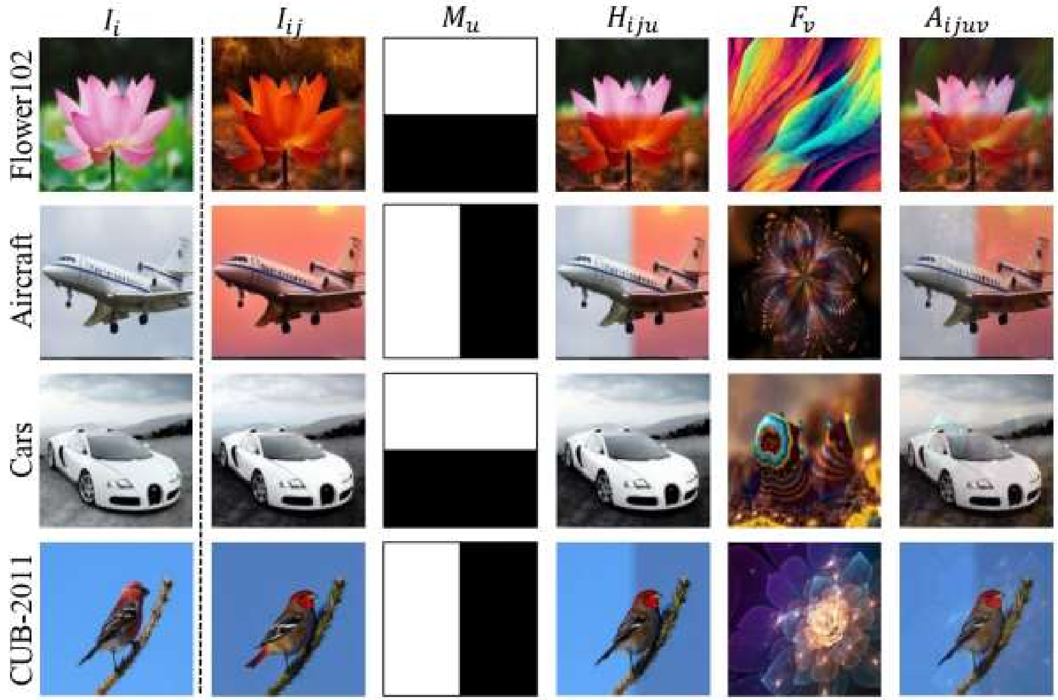


Figure 4. Example images from different stages of DIFFUSEMIX: input image (I_i), generated image ($\hat{I}_{ij}$), mask (M_u), hybrid image (H_{iju}), fractal image (F_v), and final augmented image (A_{ijuv}).

Experiments

南京航空航天大學
Nanjing University of Aeronautics and Astronautics

General Classification

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**

Adversarial Robustness

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**

Experiments

南京航空航天大學
Nanjing University of Aeronautics and Astronautics

Fine-Grained Visual 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**

Data Scarcity

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**

Transfer Learning

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**

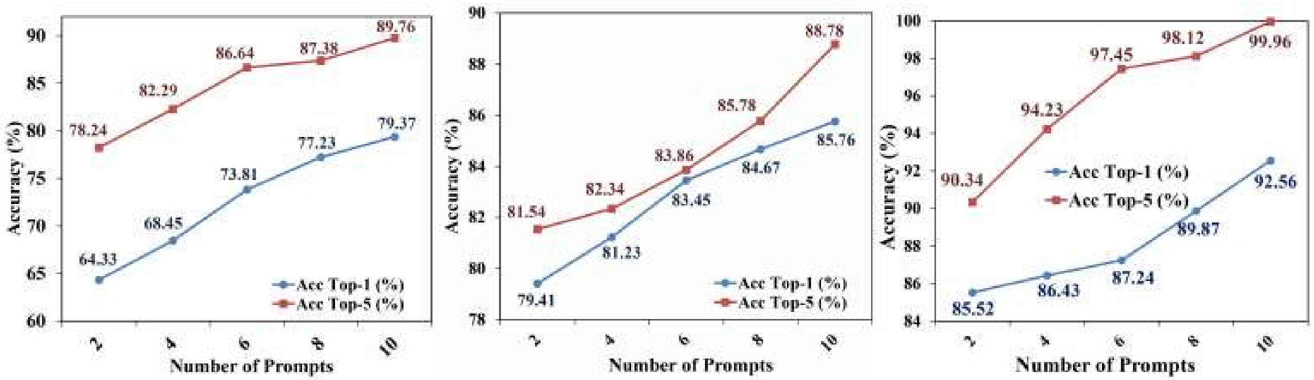
Experiments

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**

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



(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.

Thanks