

# SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data

Shaoli Huang,<sup>1</sup> Xinchao Wang,<sup>2</sup> Dacheng Tao<sup>1</sup>

<sup>1</sup> The University of Sydney

<sup>2</sup> Stevens Institute of Technology

shaoli.huang@sydney.edu.au, xinchao.wang@stevens.edu , dacheng.tao@sydney.edu.au

## Abstract

Data mixing augmentation has proved effective in training deep models. Recent methods mix labels mainly based on the mixture proportion of image pixels. As the main discriminative information of a fine-grained image usually resides in subtle regions, methods along this line are prone to heavy label noise in fine-grained recognition. We propose in this paper a novel scheme, termed as **Semantically Proportional Mixing** (SnapMix), which exploits class activation map (CAM) to lessen the label noise in augmenting fine-grained data. SnapMix generates the target label for a mixed image by estimating its intrinsic semantic composition, and allows for asymmetric mixing operations and ensures semantic correspondence between synthetic images and target labels. Experiments show that our method consistently outperforms existing mixed-based approaches on various datasets and under different network depths. Furthermore, by incorporating the mid-level features, the proposed SnapMix achieves top-level performance, demonstrating its potential to serve as a solid baseline for fine-grained recognition. Our code is available at <https://github.com/Shaoli-Huang/SnapMix.git>.

## Introduction

Despite the remarkable success of deep neural networks, its overfitting problem persists, particularly in encountering limited training data. Data Augmentation methods can alleviate this by effectively exploiting existing data. Among them, mixing-based methods (Tokozume, Ushiku, and Harada 2018; Inoue 2018; Zhang et al. 2018; Yun et al. 2019) have recently gained increasing attention. These approaches generate new data by blending images and fusing their labels according to the statistics of mixed pixels. For instance, Mixup (Zhang et al. 2018) combines images linearly and mix their targets using the same combination coefficients. CutMix (Yun et al. 2019), on the other hand, cuts out one image area, pastes it on another image, and mix their labels according to the area proportion. By extending the training distribution, mixing-based techniques reduce memorizing data and improve model generalization.

However, their superiority decreases with the increasing risk of label noise in augmenting fine-grained data. In fine-grained object recognition, discriminative information

mainly lies in some small regions of images. Mixing labels based on mixture pixel-based statistics such as area size, therefore, tends to introduce severe label noise in this task. In the example of Fig. 1, CutMix cuts out a small region covering critical information about the label, in this case a red shoulder and yellow wing bar of the red-winged blackbird. The remaining part of the image, as a result, are left with only much less informative image evidences, yet still take up a high coefficient due to its large area size. This indicates that mixing labels based on area proportion is not able to effectively reflect intrinsic semantic composition of the combined image, thereby deteriorating the data augmentation effectiveness and confusing the model training.

In this paper, we propose a novel **Semantically Proportional Mixing** (SnapMix) strategy to address this issue. SnapMix exploits a class activation map (CAM) (Zhou et al. 2016) to estimate the label composition of the mixed-images. Specifically, by normalizing the CAM of each image to sum to 1, we first obtain its Semantic Percent Map (SPM) to quantify the relatedness percentage between each pixel and the label, and then compute the semantic ratio of any image region by summing values in the corresponding area of the SPM. For an image composed of multiple areas from multiple images, we can estimate its semantic composition through the semantic ratios corresponding to these regions. Compared to methods based on statistics of mixture pixels, our label mixing strategy incorporates neural activations as prior knowledge to ensure the semantic correspondence between the synthetic images and the generated supervision signals.

Moreover, existing techniques rely on *symmetrically* blending image regions, meaning that the selected areas to be mixed are restricted to be complementary, and hence limit the diversity of augmented data. By contrast, the proposed approach enables *asymmetric* cut-and-paste operations, allowing us to incorporate various factors such as deformation and scale into the data augmentation to boost the data diversity. The current label-mixing strategies are designed based on the complementary principle. Thus they are not suitable for the *asymmetric* operation that selects non-complementary regions to mix.

To validate the proposed approach, we adopt various network architectures (Resnet-18,34,50,101 (He et al. 2016)) as baseline models and compare our method with existing data

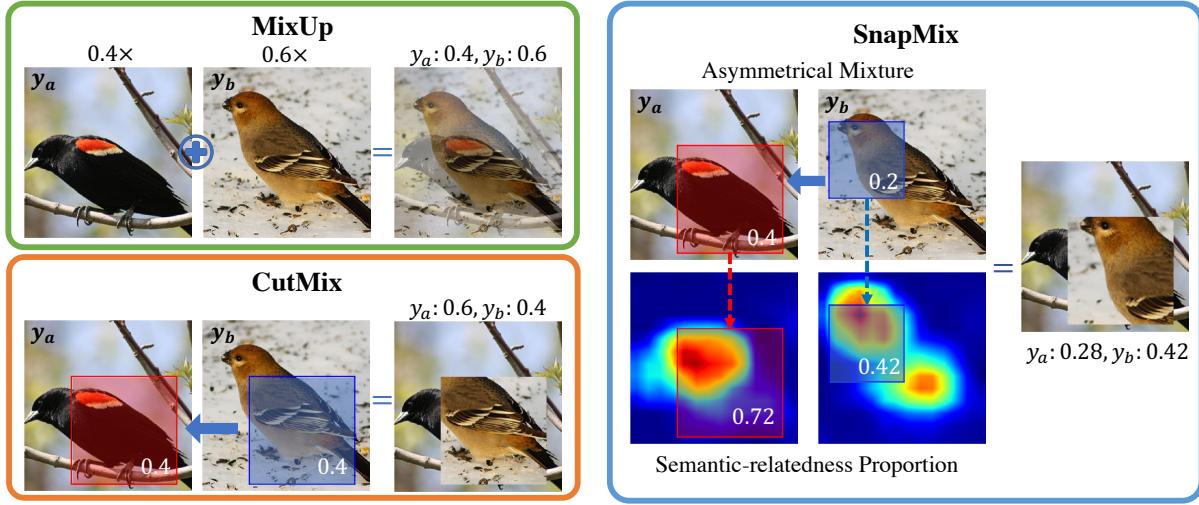


Figure 1: Comparison of Mixup, CutMix, and SnapMix. The figure gives an example where SnapMix’s generated label is visually more consistent with the mixed image’s semantic structure comparing to CutMix and Mixup.

augmentation approaches on three fine-grained datasets. Results indicate that prior methods lead to unstable performances, sometimes even harmful, when using shallow network architecture. This can be in part explained by the fact shallow neural networks are not able to well tackle label noises, which significantly degrade data augmentation effectiveness. The proposed method, on the other hand, consistently outperforms compared methods on various datasets and with different network depths. Furthermore, we show that even a simple model can achieve comparable state-of-the-art performance when applying our proposed data augmentation. This indicates that our method can well serve as a solid baseline for advancing fine-grained recognition.

## Related Works

### Fine-Grained Classification

Fine-grained recognition has been an active research area in recent years. This task is more challenging than general image classification (Liu and Tao 2016; Wang, Li, and Tao 2011; Yang et al. 2018; Yang et al. 2020b,a), as the critical information to distinguish categories usually lies in subtle object parts. Part-based methods thereby, are extensively explored to address the problem. Early works (Huang et al. 2016; Zhang et al. 2014; Xiao et al. 2015; Lin et al. 2015; Xu et al. 2015, 2016) mainly rely on strongly supervised learning to localize object part for subsequent feature learning. Due to part annotations are expensive to acquire, the later methods (Zhang et al. 2016; Zheng et al. 2017; Sun et al. 2018; Zheng et al. 2019) attempts to find discriminative part regions in a weakly supervised manner. For example, Zhang et al. (Zhang et al. 2016) first picks distinctive filters and then use them to learn part detectors through an iteratively alternating strategy. MA-CNN (Zheng et al. 2017) obtains part regions by clustering feature maps of intermediate convolutional layers, MAMC (Sun et al. 2018). In recent years, fine-grained approaches have developed in the direction of

enforcing the neural networks to acquire rich information (Yang et al. 2018; Ding et al. 2019; Chen et al. 2019; Zhang et al. 2019). For instance, Zhang et al.,(Zhang et al. 2019) progressively crop out discriminative regions to generate diversified data sets for training network experts. Chen et al., (Chen et al. 2019) destruct the images and then learn a region alignment network to restore the original spatial layout of local regions. These works implicitly integrate data augmentation practices into their methodologies, which relate to our proposed method mostly.

However, our proposed method SnapMix differs from them in two aspects. First, SnapMix is a pure data augmentation based technique that does not require an extra computational process in the testing stage. Besides, our approach builds on recent advances from the data mixing strategy. In contrast, those methods are mainly based on conventional data augmentation strategy, which typically processes a single image and retains the original label.

### Data augmentation

Recent advances (Takahashi, Matsubara, and Uehara 2019; Zhong et al. 2017; DeVries and Taylor 2017; Tokozume, Ushiku, and Harada 2018; Inoue 2018; Zhang et al. 2018; Yun et al. 2019) in data augmentation can be divided into two groups: region-erasing based and data mixing. The former (Zhong et al. 2017; DeVries and Taylor 2017) erases partial region of images in training, aiming to encourage the neural networks to find more discriminative regions. The typical method is CutOut that augments data by cutting a rectangle region out of an image. The other line of methods is data mixing based (Tokozume, Ushiku, and Harada 2018; Inoue 2018; Zhang et al. 2018) that have recently gained increasing attention in the field of image classification. Compared with region-erasing augmentation, these methods generate new data by combining multiple images and fusing their labels accordingly. Among those works, Zhang

et al., (Zhang et al. 2018) first proposed mixing data to extend the training distribution. Their proposed method termed as MixUp, generated images by linearly combining images and fusing their labels using the same coefficients. MixUp showed its superiorities in handling corrupted targets and improving model performance. Summers and Dineen (Summers and Dineen 2019) further improved Mixup by introducing a more generalized form of data mixing that considered non-linear mixing operations. In very recent work, Yun et al. proposed CutMix (Yun et al. 2019) that produces a new image by cutting out one image patch and pasting to another image. Similar to Mixup, the labels are also mixed but proportionally to the area of the patches. By taking advantage of both types of methods, CutMix showed impressive performance in classification tasks and weakly-supervised localization tasks.

Our proposed method falls into the second category. However, it differs significantly from the previous techniques in the way of mixing labels. Current mixing-data based approaches combine labels mainly depending on the statistic of mixture pixels, such as the ratio of pixel number or intensity values. In comparison, our method estimates the semantic structure of a synthetic image by exploiting class activation maps. This new characteristic allows our approach to augment fine-grained data without introducing severe label noise. Another slight difference is that SnapMix blends images using asymmetric patches, resulting in better data randomness and diversity than those using symmetric regions.

### Semantically Proportional Mixing

Data augmentation has become an indispensable step for training deep neural networks. The standard augmentation methods mainly apply a composition of image preprocessing techniques on an input image, such as flipping, rotations, color jittering, and random cropping. Recent works demonstrated the great potential of mixing-based techniques for training deep models. Unlike standard practice, these methods generate new data by combining images and also mixing the corresponding labels. In the following, we first provide some notations used in this paper. We then briefly introduce two representative mix-based approaches Mixup and CutMix. Next, we describe in detail our proposed method SnapMix.

### Notations

We use the following notations throughout this paper. The original training data set  $\{(I_i, y_i) | i \in [0, 1, \dots, N - 1]\}$ , where  $I_i \in R^{3 \times W \times H}$  and  $y_i$  refer to an input image and the label respectively. Given a data pair  $((I_a, y_a), (I_b, y_b))$  and hyperparameter  $\alpha$ , mixing-based methods first draw a random value  $\lambda$  from a beta distribution  $Beta(\alpha, \alpha)$ . Then they generate a new image  $\tilde{I}$  and two label weights  $\rho_a$  and  $\rho_b$  according to  $\lambda$ . Here,  $\rho_a$  and  $\rho_b$  are corresponding to the label  $y_a$  and  $y_b$  respectively.

### Mixup and CutMix

Recent mixing-based methods essentially stem from two representative techniques Mixup and cutMix.

*MixUp* mixes images and combines labels using linear combination, which is expressed as

$$\begin{aligned} \tilde{I} &= \lambda \times I_a + (1 - \lambda) \times I_b, \\ \rho_a &= \lambda, \rho_b = 1 - \lambda, \end{aligned} \quad (1)$$

*CutMix* adopts cut-and-paste operation for mixing images and mixes the labels according to the area ratio. That is

$$\begin{aligned} \tilde{I} &= (1 - M_\lambda) \odot I_a + M_\lambda \odot I_b, \\ \rho_a &= 1 - \lambda, \rho_b = \lambda, \end{aligned} \quad (2)$$

where  $\odot$  denotes element-wise multiplication and  $M_\lambda \in R^{W \times H}$  is a binary mask of a random box region whose area ratio to the image is  $\lambda$ .

On one hand, these two methods mainly differ in the way they mix images. Mixup mixes image by linear combination and therefore improves the neural networks' robustness to adversarial examples. By integrating Mixup and regional dropout strategy, the cut-and-paste regime of Cutmix naturally inherits this advantage but also enhances models' capabilities of object localization. On the other hand, they share two similarities: 1) mixing labels by using the statistic of mixture pixels, and 2) performing image mixing in symmetric locations.

### SnapMix for Fine-grained Recognition

In fine-grained recognition, the category difference usually resides in subtle object parts, making part localization ability plays an important role. Therefore, the cut-and-paste mechanism is more favorable in augmenting fine-grained data. However, mixing labels by the region area ratio is unreliable and will increase the risk of label noise, particularly in combining image at asymmetric locations. Motivated by work that used class activation maps (CAMs) to describe the class-specific discriminative regions, we propose to exploit CAMs to estimate the semantic composition of a mixed image. Fig. 2 shows an overview of our proposed method SnapMix. Our proposed method differs existing methods in two folds: 1) fusing labels based on semantic composition estimation, 2) mixing images asymmetrically. Given an input pair of data, we first extract their semantic percentage maps used to compute the semantic percentage of any image area. We then mix the images by cut-and-paste at asymmetrical locations. Finally, we calculate each mixture component's semantic proportion as guidance to fuse the one-hot labels. In the following, we further describe in detail our method in terms of image mixing and label generation.

**Mixing images.** As discussed previously, current existing methods blend images at symmetric locations, limiting the diversity of synthetic images. Our approach removes this constrain to increase the randomness of data augmentation further. Specifically, instead of using a single random location, we crop an area at a random location in one image and transform and paste it to another random place in another image. Such mixing operation is expressed as

$$\tilde{I} = (1 - M_{\lambda^a}) \odot I_a + T_\theta(M_{\lambda^b} \odot I_b), \quad (3)$$

where  $M_{\lambda^a}$  and  $M_{\lambda^b}$  are two binary masks containing random box regions with the area ratios  $\lambda^a$  and  $\lambda^b$ , and  $T_\theta$  is a

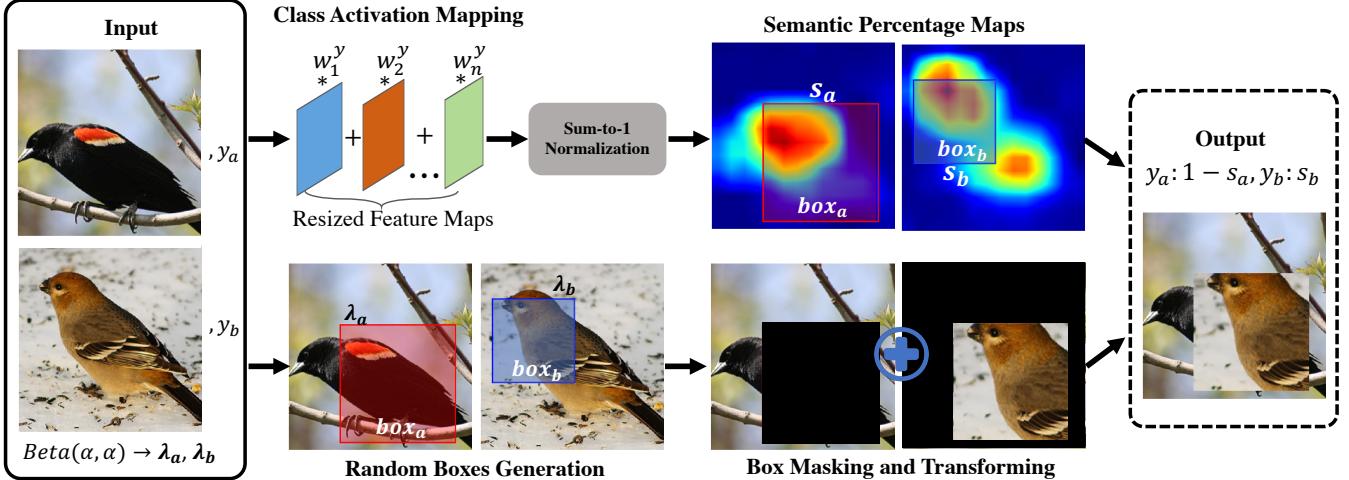


Figure 2: An overview of proposed method.

function that transforms the cutout region of  $I_b$  to match the box region of  $I_a$ .

**Label generation.** To estimate the semantic composition of a mixed image, we need to measure each original image pixel's semantic relatedness to the corresponding label. One alternative to do this can resort to class activation map, as it is proved useful to interpret how a region correlates with a semantic class. Thus, we first employ the attention method(Zhou et al. 2016) to compute the class activation maps of input images. For a given image  $I_i$ , we denote  $F(I_i) \in \mathbb{R}^{d \times h \times w}$  the output of the last convolutional layer,  $F_l(I_i)$  the  $l^{th}$  feature map of  $F(I_i)$ , and  $w_{y_i} \in \mathbb{R}^d$  the classifier weight corresponding to class  $y_i$ . Then we can obtain  $I_i$ 's class activation map  $CAM(I_i)$  by

$$CAM(I_i) = \Phi\left(\sum_{l=0}^d w_{y_i}^l F_l(I_i)\right), \quad (4)$$

where  $\Phi(\cdot)$  denotes a operation that upsamples a feature map to match dimensions with input image size. Here, we ignore the bias term for simplicity. We can now obtain a Semantic Percent Map (SPM) by normalizing the CAM to sum to one. Here, we define SPM as a semantic information measure map to quantify the relatedness percentage between a pixel and the label. We compute the SPM of an image  $S(I_i)$  by

$$S(I_i) = \frac{CAM(I_i)}{\text{sum}(CAM(I_i))}, \quad (5)$$

Finally, for an image produced using Eq.3, we compute the corresponding label weights  $\rho_a$  and  $\rho_b$  as

$$\begin{aligned} \rho_a &= 1 - \text{sum}(M_{\lambda^a} \odot S(I_a)), \\ \rho_b &= \text{sum}(M_{\lambda^b} \odot S(I_b)). \end{aligned} \quad (6)$$

By doing so, the generated supervision information for a mixed image can better reflect its intrinsic semantic composition. Therefore, in fine-grained recognition, despite the

image's discriminative information is extremely uneven in spatial distribution, our method prevent introducing heavy noise in the augmented data.

It is also worth noting that the two components of a mixed image generally do not complement each other in terms of semantic proportion. A case of this would be when a cutout is a background patch and pasted over the object area of another image, and then the synthesized image would not contain any foreground object. Therefore, unlike CutMix, our method does not restrict the label coefficients ( $\rho_a$  and  $\rho_b$ ) to sum up to 1.

## Experiments

In this section, we extensively evaluated the performance of SnapMix on three fine-grained datasets. We evaluated our method using multiple network structures (Resnet-18,34,50,101) as baselines. We compared the performance of our approach and related data augmentation methods on each network architecture. Further, we tested our method using a strong baseline that integrated mid-level features and compared the results with those of the current state-of-the-art methods of fine-grained recognition.

## Datasets

We conduct experiments on three standard fine-grained datasets, which are CUB-200-2011 (Wah et al. 2011), Stanford-Cars (Krause et al. 2013), and FGVC-Aircraft (Maji et al. 2013). For each dataset, We first resized images to  $512 \times 512$  and cropped them with size  $448 \times 448$ . In the rest of the paper, we used the short names CUB, Cars, and Aircraft to simplify the notation.

## Experiment Setup

**Backbone networks and baselines.** To extensively compare our method with other approaches, we used four network backbones as baselines in performance comparison. Here, if not specified, we refer baseline as a neural network

Table 1: Performance comparison(Mean Acc.%) of methods using backbone networks *Resnet-18* and *Resnet-34* on fine-grained datasets. Each method’s improvement over the baseline is shown in the brackets.

	CUB		Cars		Aircraft	
	Res18	Res34	Res18	Res34	Res18	Res34
Baseline	82.35	84.98	91.15	92.02	87.80	89.92
CutOut	80.54 (-1.81)	83.36 (-1.62)	91.83(+0.68)	92.84 (+0.82)	88.58 (+0.78)	89.90 (-0.02)
MixUp	83.17 (+0.82)	85.22 (+0.24)	91.57 (+0.42)	93.28 (+1.26)	89.82 (+2.02)	91.02 (+1.1)
CutMix	80.16 (-2.19)	85.69 (+0.71)	92.65 (+1.50)	93.61 (+1.59)	89.44 (+1.64)	91.26 (+1.34)
SnapMix	<b>84.29 (+1.94)</b>	<b>87.06 (+2.08)</b>	<b>93.12(+1.97)</b>	<b>93.95 (+1.93)</b>	<b>90.17 (+2.37)</b>	<b>92.36 (+2.44)</b>

Table 2: Performance comparison(Mean Acc.%) of methods using backbone networks *Resnet-50* and *Resnet-101* on fine-grained datasets. Each method’s improvement over the baseline is shown in the brackets.

	CUB		Cars		Aircraft	
	Res50	Res101	Res50	Res101	Res50	Res101
Baseline	85.49	85.62	93.04	93.09	91.07	91.59
CutOut	83.55 (-1.94)	84.70 (-0.92)	93.76 (+0.72)	94.16 (+1.07)	91.23 (+0.16)	91.79 (+0.2)
MixUp	86.23 (+0.74)	87.72 (+2.1)	93.96 (+0.92)	94.22 (+1.13)	92.24 (+1.17)	92.89 (+1.3)
CutMix	86.15 (+0.66)	87.92 (+2.3)	94.18 (+1.14)	94.27 (+1.18)	92.23 (+1.16)	92.29 (+0.7)
SnapMix	<b>87.75 (+2.26)</b>	<b>88.45 (+2.83)</b>	<b>94.30 (+1.21)</b>	<b>94.44 (+1.35)</b>	<b>92.80 (+1.73)</b>	<b>93.74 (+2.15)</b>

model that was pre-trained on Imagenet dataset and fine-tuned on a target dataset. The used network structures include Resnet-18,34,50 and 101. Here, we adapted their implementation from the TorchVision package to our experiments.

We also used a strong baseline that incorporates mid-level features in performance evaluation. Here, we termed it **baseline<sup>†</sup>**. This baseline was used in recent works(Wang, Morariu, and Davis 2018; Zhang et al. 2019) to push the performance limits of fine-grained recognition. Compared with the standard baseline that contains a single classification branch, Baseline<sup>†</sup> adds another mid-level classification branch on top of the intermediate layers. In our experiments, we followed the implementation from (Zhang et al. 2019). Specifically, the mid-level branch included a *Conv1×1*, *Max Pooling*, and a *Linear Classifier layer* and was placed after 4<sup>th</sup> block of ResNet. We blocked the gradients passing the mid-level branch to backbone networks in training. In testing, we fused the predictions from two classification branches.

**Data augmentation methods.** We compared our method with three representative data augmentation methods namely CutOut (DeVries and Taylor 2017), MixUp (Zhang et al. 2018), and CutMix (Yun et al. 2019). Since these works did not officially report results on fine-grained datasets, we implemented these methods based on the released codes and run experiments on fine-grained datasets. We first tested different hyperparameters for each method and then selected the optimal one for all network structures. We set the probability of performing augmentation 0.5 for CutOut and MixUp and 1.0 for CutMix. We used the  $\alpha$  values of 1.0 and 3.0 for MixUp and CutMix, respectively.

**Training details.** We used stochastic gradient descent (SGD) with momentum 0.9, base learning rate 0.001 for the pre-trained weights, and 0.01 for new parameters. We trained our model for 200 epochs and decayed the learning rate by factor 0.1 every 80 epochs.

## Performance evaluation

In this section, we presented the results of our method and performance comparisons with existing approaches. We first made comparisons between SnapMix and other data augmentation methods. Further, we tested our approach using the two baselines and compared the results with those of the current state-of-the-art methods. We used top-1 accuracy as the performance measure and provided both the best accuracy and average accuracy (the mean result of the final 10 epochs) of our proposed method.

**Comparison with data augmentation methods.** We listed the results of performance comparisons in Table. 1-2. Here, Table 1-2 shows each method’s average accuracy and improvement over the baseline. First, we can observe that our proposed method SnapMix consistently outperforms its counterparts on three datasets. We can further find that existing methods mostly yield limited, even negative improvement on the CUB dataset. This might mainly because the CUB dataset exhibits more subtle category differences, making those methods increase the risk of noise labels. Besides, the effectiveness of these methods is relatively sensitive to the network depth. For example, both Mixup and CutMix achieve significant improvements on the CUB dataset only using the deeper networks Resnet-101, and CutMix even suffers a performance drop when using Resnet-18. We hypothesis the reason is that the deeper models have better ca-

Method	Backbone	Accuracy(%)		
		CUB	Cars	Aircraft
RA-CNN (Fu, Zheng, and Mei 2017)	$3 \times$ VGG-19	85.3	92.5	88.2
RAM (Li et al. 2017)	$3 \times$ Res-50	86.0	-	
Kernel-Pooling (Cui et al. 2017)	$1 \times$ VGG-16	86.2	92.4	86.9
NTS-Net (Yang et al. 2018)	$5 \times$ Res-50	87.5	93.9	91.4
DFL-CNN (Wang, Morariu, and Davis 2018)	$1 \times$ VGG-16	86.7	93.8	92.0
MAMC (Sun et al. 2018)	$1 \times$ Res-101	86.5	93.0	-
DFL-CNN (Wang, Morariu, and Davis 2018)	$1 \times$ Res-50	87.4	93.1	91.7
DCL (Chen et al. 2019)	$1 \times$ Res-50	87.8	94.5	93.0
TASN (Zheng et al. 2019)	$1 \times$ Res-50	87.9	93.8	-
S3N (Ding et al. 2019)	$3 \times$ Res-50	88.5	94.7	92.8
MGN-CNN (Zhang et al. 2019)	$3 \times$ Res-50	88.5	93.9	-
MGN-CNN (Zhang et al. 2019)	$3 \times$ Res-101	<b>89.4</b>	93.6	-
baseline	$1 \times$ Res-50	85.49 (85.85)	93.04 (93.17)	91.07 (91.30)
baseline <sup>†</sup>	$1 \times$ Res-50	87.13	93.80	91.68
baseline	$1 \times$ Res-101	85.62 (86.02)	93.09 (93.28)	91.59 (92.11)
baseline <sup>†</sup>	$1 \times$ Res-101	87.81	93.94	91.85
baseline + SnapMix	$1 \times$ Res-50	87.75 (88.01)	94.30 (94.59)	92.80 (93.16)
baseline <sup>†</sup> + SnapMix	$1 \times$ Res-50	88.70 (88.97)	<b>95.00</b> (95.16)	93.24(93.49)
baseline + SnapMix	$1 \times$ Res-101	88.45 (88.73)	94.44 (94.60)	93.74 (94.03)
baseline <sup>†</sup> + SnapMix	$1 \times$ Res-101	89.32 (89.58)	94.84 (94.96)	<b>94.05</b> (94.24)

Table 3: The accuracy (%) comparison with state-of-the-art methods on CUB, Cars, and Aircraft. For the baselines and our approach, we reported their average accuracy of the final ten epochs and showed their best accuracy in the brackets.

pacities in handling label noise. In comparison, SnapMix significantly improves the baseline regardless of network depth.

#### Comparison with state-of-the-art methods.

In this section, we compared the performance of SnapMix and other state-of-the-art techniques of fine-grained recognition. In Table. 3, first, we can observe that the baseline<sup>†</sup> achieved higher accuracy than the baseline on three datasets, and the performance gain on the CUB dataset is the most significant. This result indicates mid-level features can effectively complement the capacity of the global-level features in fine-grained recognition. It is also worth mentioning that some top-performing works, such as DFL-CNN (Wang, Morariu, and Davis 2018) and MGN-CNN (Zhang et al. 2019) also embedded the baseline+ into their methods.

Secondly, SnapMix enhances both baselines to obtain comparable performance even to some latest approaches with intricate designs and high inference time. S3N (Ding et al. 2019) and MGN-CNN (Zhang et al. 2019) are two of the state-of-the-art methods. S3N adopted a selective sparse sampling strategy to construct multiple features. MGN-CNN exploit attention mechanisms to construct different inputs for multiple expert networks. Both methods require a similar data processing pipeline with the training stage and the need for multiple feed-forward passes of the backbone network in the testing stage.

In contrast, using a standard baseline with a single Resnet-101 backbone, SnapMix, without bells and whistles in the testing stage, achieves the accuracy of 88.45%, 94.44%, and

93.74% on CUB, Cars, and Aircraft respectively, which outperforms most of the existing techniques. Even using the baseline<sup>†</sup> (a more powerful baseline), SnapMix still demonstrates its promise and effectiveness in performance improvement, pushing the accuracy to the next level. For example, SnapMix achieves 89.32% accuracy (close to the result of MGN-CNN 89.4%) on the CUB dataset and exhibits superior performance than all the comparing techniques on both Cars and Aircraft dataset.

#### Analysis

**Training from scratch.** Tab. 4 shows that our approach is also effective without using ImageNet pre-trained weights. In this experiment, we used the 'switch' probability (the probability of applying the mixing augmentation) of 0.5 for each mixing method. This allows the networks to learn from both clean and mixed data, preventing the mixed data from excessively affecting the model's initial learning stage. Therefore, despite SnapMix may introduce noise labels in the early training stage, it would not hinder the network from learning a good CAM in the subsequent stage. This is because the network tends to first learn from easy samples (clean data ) other than difficult samples (mixed data with label noise) (Arpit et al. 2017). With the continuous learning of the network and the improvement of CAM quality, the more reasonable the label estimated by SnapMix will be to enhance subsequent model learning.

**Effectiveness of using other network backbones.** We evaluated the performance of our method with two other network

Table 4: Performance comparison of training from scratch on the CUB dataset (Acc.%).

	Baseline	CutMix	MixUp	SnapMix
Res-18	64.98	60.03	67.63	<b>70.31</b>
Res-50	66.92	65.28	<b>72.39</b>	72.17

Table 5: Performance comparison of using other network backbones on the CUB dataset (Acc.%).

	Baseline	Cutmix	Mixup	Snapmix
InceptionV3	82.22	84.31	83.83	<b>85.54</b>
DenseNet121	84.23	86.11	86.65	<b>87.42</b>

backbones including InceptionV3 (Szegedy et al. 2016) and DenseNet121 (Huang et al. 2017). As shown in Tab 5, our method surpasses both CutMix and MixUp approaches and improves the baseline by a large margin. This result demonstrates SnapMix’s consistent effectiveness when applied to various CNN architecture.

**Influence of hyperparameters.** The hyperparameter  $\alpha$  of snapMix decides a beta distribution that is used to generate a random patch in mixing. To investigate its impact on the performance, we tested seven values of  $\alpha$ . Table.6 showed that the accuracy increased slightly with the increase of  $\alpha$  value and peaked at the number of 5, which suggests the importance of using the medium-size boxes to mix images on this dataset. Besides, the accuracy of setting different  $\alpha$  values inconsiderably fluctuates around the mean value of 87.37%, indicating that snapMix is not very sensitive to the  $\alpha$  value.

**Effectiveness of each component of SnapMix.** We performed experiments using combinations of different image mixing operations and label mixing strategies. As shown in Fig. 3, the asymmetric mixing provides a slight improvement over the symmetric mixing, and the label mixing strategy of SnapMix is the primary contributor to the performance gain. More importantly, the Semantic-Ratio consistently shows improvement in using three image mixing operations.

**Visualization.** Fig. 4 shows CAMs of some examples correctly predicted by SnapMix but misclassified by MixUp and CutMix. We can observe the attention of MixUp and CutMix are distracted by some background patterns, which might be a reason for the misprediction. By comparison, the network attention of SnapMix tends to lie in object regions.

Table 6: Influence of hyperparameters Acc.(%)

$\alpha=0.2$	$0.5$	$1.0$	$3.0$	$5.0$	$7$	$8$
87.22	87.23	87.25	87.30	87.75	87.30	87.54

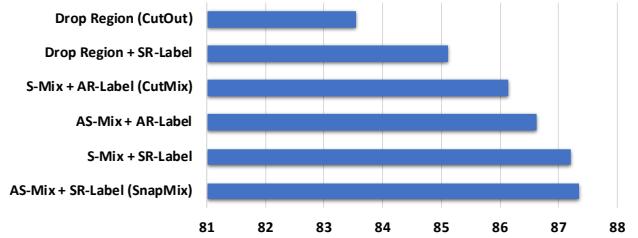


Figure 3: Accuracy comparison of six different combination techniques (%). Here, **S-Mixing**, **AS-Mix**, **AR-label**, and **SR-label** are short for symmetric mixing, asymmetric mixing, area ratio label, and semantic ratio label respectively.

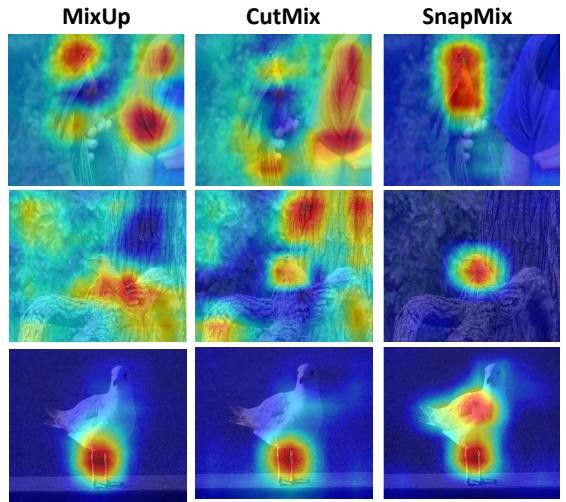


Figure 4: CAM visualization of different augmentation methods.

These results imply mixing labels by pixel statistics may cause the neural networks more sensitive to background visual patterns, while our proposed method avoids this issue.

## Conclusions

In this paper, we present a new method SnapMix for augmenting fine-grained data. SnapMix generates new training data with more reasonable supervision signals by considering the semantic correspondence. Our experiments showed the importance of estimating semantic composition for a synthetic image. Our proposed method might also benefit other tasks (e.g., indoor scene recognition or person re-identification), where a small image region contains significant discriminative information. The proposed label mixing strategy is mainly applicable to cut-and-paste mixing. Further work might explore how better to estimate the semantic structure of a linearly combined image.

## Acknowledgements

This work was supported by the Australian Laureate Fellowship project FL170100117, DP-180103424, IH-180100002, and Stevens Institute of Technology Startup Funding.

## References

- Arpit, D.; Jastrzebski, S. K.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A. C.; Bengio, Y.; et al. 2017. A Closer Look at Memorization in Deep Networks. In *ICML*.
- Chen, Y.; Bai, Y.; Zhang, W.; and Mei, T. 2019. Destruction and Construction Learning for Fine-grained Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5157–5166.
- Cui, Y.; Zhou, F.; Wang, J.; Liu, X.; Lin, Y.; and Belongie, S. 2017. Kernel pooling for convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* .
- Ding, Y.; Zhou, Y.; Zhu, Y.; Ye, Q.; and Jiao, J. 2019. Selective Sparse Sampling for Fine-Grained Image Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 6599–6608.
- Fu, J.; Zheng, H.; and Mei, T. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *IEEE conference on computer vision and pattern recognition*, 4438–4446.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.
- Huang, S.; Xu, Z.; Tao, D.; and Zhang, Y. 2016. Part-stacked cnn for fine-grained visual categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1173–1182.
- Inoue, H. 2018. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929* .
- Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 554–561.
- Li, Z.; Yang, Y.; Liu, X.; Zhou, F.; Wen, S.; and Xu, W. 2017. Dynamic Computational Time for Visual Attention.
- Lin, D.; Shen, X.; Lu, C.; and Jia, J. 2015. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1666–1674.
- Liu, T.; and Tao, D. 2016. Classification with Noisy Labels by Importance Reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(3): 447–461.
- Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151* .
- Summers, C.; and Dinneen, M. J. 2019. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1262–1270. IEEE.
- Sun, M.; Yuan, Y.; Zhou, F.; and Ding, E. 2018. Multi-Attention Multi-Class Constraint for Fine-grained Image Recognition. *European Conference on Computer Vision* .
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.
- Takahashi, R.; Matsubara, T.; and Uehara, K. 2019. Data augmentation using random image cropping and patching for deep CNNs. *IEEE Transactions on Circuits and Systems for Video Technology* .
- Tokozume, Y.; Ushiku, Y.; and Harada, T. 2018. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5486–5494.
- Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset .
- Wang, X.; Li, Z.; and Tao, D. 2011. Subspaces Indexing Model on Grassmann Manifold for Image Search. *IEEE Transactions on Image Processing* 20(9): 2627–2635.
- Wang, Y.; Morariu, V. I.; and Davis, L. S. 2018. Learning a discriminative filter bank within a CNN for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4148–4157.
- Xiao, T.; Xu, Y.; Yang, K.; Zhang, J.; Peng, Y.; and Zhang, Z. 2015. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 842–850.
- Xu, Z.; Huang, S.; Zhang, Y.; and Tao, D. 2015. Augmenting strong supervision using web data for fine-grained categorization. In *IEEE International Conference on Computer Vision*, 2524–2532.
- Xu, Z.; Tao, D.; Huang, S.; and Zhang, Y. 2016. Friend or foe: Fine-grained categorization with weak supervision. *IEEE Transactions on Image Processing* 26(1): 135–146.
- Yang, E.; Deng, C.; Li, C.; Liu, W.; Li, J.; and Tao, D. 2018. Shared Predictive Cross-Modal Deep Quantization. *IEEE Transactions on Neural Networks and Learning Systems* 29(11): 5292–5303.
- Yang, Y.; Feng, Z.; Song, M.; and Wang, X. 2020a. Factorizable Graph Convolutional Networks. *Advances in Neural Information Processing Systems* 33.
- Yang, Y.; Qiu, J.; Song, M.; Tao, D.; and Wang, X. 2020b. Distilling Knowledge From Graph Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Yang, Z.; Luo, T.; Wang, D.; Hu, Z.; Gao, J.; and Wang, L. 2018. Learning to Navigate for Fine-grained Classification. In *European Conference on Computer Vision*, 420–435.

- Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.
- Zhang, L.; Huang, S.; Liu, W.; and Tao, D. 2019. Learning a Mixture of Granularity-Specific Experts for Fine-Grained Categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, 8331–8340.
- Zhang, N.; Donahue, J.; Girshick, R.; and Darrell, T. 2014. Part-based R-CNNs for fine-grained category detection. In *European conference on computer vision*, 834–849. Springer.
- Zhang, X.; Xiong, H.; Zhou, W.; Lin, W.; and Tian, Q. 2016. Picking deep filter responses for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1134–1142.
- Zheng, H.; Fu, J.; Mei, T.; and Luo, J. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE international conference on computer vision*, 5209–5217.
- Zheng, H.; Fu, J.; Zha, Z.-J.; and Luo, J. 2019. Looking for the Devil in the Details: Learning Trilinear Attention Sampling Network for Fine-grained Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5012–5021.
- Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2017. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896* .
- Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2921–2929.