ResizeMix: Mixing Data with Preserved Object Information and True Labels

Jie Qin[1,2*], Jiemin Fang[3,4*], Qian Zhang[5], Wenyu Liu[4], Xingang Wang[2†], Xinggang Wang[4]

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]Institute of Automation, Chinese Academy of Sciences
[3]Institute of Artificial Intelligence, Huazhong University of Science and Technology
[4]School of EIC, Huazhong University of Science and Technology [5]Horizon Robotics
{qinjie2019, xingang.wang}@ia.ac.cn qian01.zhang@horizon.ai
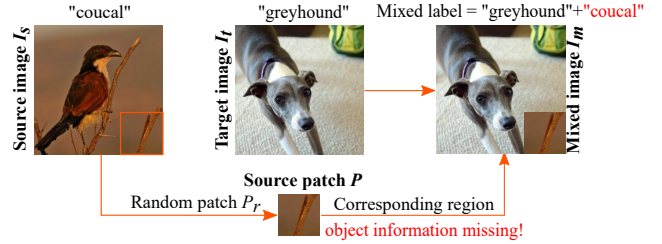{jaminfong, liuwy, xgwang}@hust.edu.cn

Abstract

*Data augmentation is a powerful technique to increase the diversity of data, which can effectively improve the generalization ability of neural networks in image recognition tasks. Recent data mixing based augmentation strategies have achieved great success. Especially, CutMix uses a simple but effective method to improve the classifiers by randomly cropping a patch from one image and pasting it on another image. To further promote the performance of CutMix, a series of works explore to use the saliency information of the image to guide the mixing. We systematically study the importance of the saliency information for mixing data, and find that the saliency information is not so necessary for promoting the augmentation performance. Furthermore, we find that the cutting based data mixing methods carry two problems of **label misallocation** and **object information missing**, which cannot be resolved simultaneously. We propose a more effective but very easily implemented method, namely ResizeMix. We mix the data by directly resizing the source image to a small patch and paste it on another image. The obtained patch preserves more substantial object information compared with conventional cut-based methods. ResizeMix shows evident advantages over CutMix and the saliency-guided methods on both image classification and object detection tasks without additional computation cost, which even outperforms most costly search-based automatic augmentation methods.*
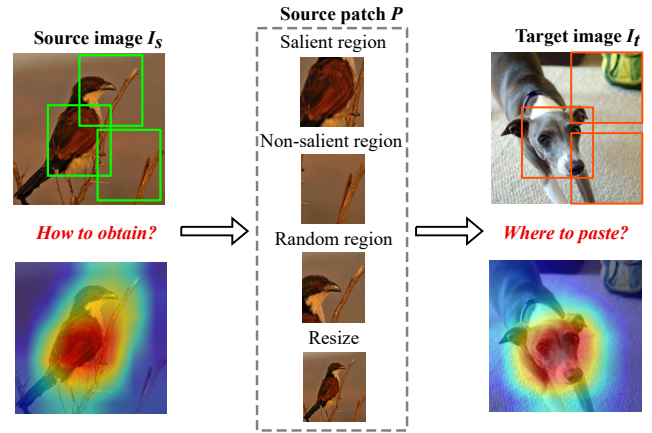
1. Introduction

Deep convolutional neural networks (CNN) have achieved great success in a wide range of computer vi-

(a) CutMix with Object Information Missing and Label Misallocation

(b) Possible Choices of Data Mixing

Figure 1. (a) illustrates that in CutMix [51], there exist two issues of object information missing and label misallocation. (b) represents different cropping manners from the source image and different pasting manners to the target image. We systematically check the two problems of "how to obtain the patch" and "where to paste the patch".

sion applications, *e.g.*, image classification [24, 44], object detection [36, 41], and semantic segmentation [6, 42] *etc*. Recent optimization techniques have further promoted the CNN performance to a new level, including data augmentation [8, 51], optimizer design [1, 30], learning rate sched-

---

ule [39], and hyper-parameter optimizing [3, 12] *etc*. Notably, the strategy of data augmentation plays a critically important role in broadening the distribution of data, which facilitates the generalization ability of the trained CNN, and effectively promotes the final performance. Advanced data augmentation methods have been widely explored for training stronger neural networks.

A series of data augmentation methods [48, 51, 53] aim at mixing data to increase the data diversity. Meanwhile, the mixed data forces the network to pay attention to multiple objects and locations in the input image, which strengthens the feature extraction ability of the networks. Especially, as shown in Fig. 1(a), CutMix [51] achieves very promising results on image classification by randomly cutting a *source patch* from a *source image* and pasting it on the *target image* at the same location. The ground truth labels are accordingly mixed proportionally to the patch area, which leads to a multi-label training style. However, the mixing strategy with a random manner may mislead the training, as the cropped patch usually does not conform to the label of the whole image. Subsequent researches [22, 29, 50] make efforts to mix the data more precisely, most of which take full advantages of the saliency information and use the location of the salient regions in the image to guide the mixing. The saliency-guided method facilitates the consistency of the mixed data and the allocated ground truth labels. This alleviates the misleading caused by the random mixing strategy during training, and further promotes the neural network performance.

However, the procedure of locating the salient region of the image always requires a complicated module and introduces additional computation cost during training, *e.g*., PuzzleMix [29] proposes to optimize the mixing mask and the saliency discounted transportation, and SaliencyMix [50] uses a saliency detection module to select the saliency source patch for mixing. In this paper, we systematically check the importance of the image saliency information for data mixing during network training. As shown in Fig. 1(b), the checking is performed mainly from two aspects, *i.e*. whether the saliency information is necessary for determining *(i) where to paste the source patch* and *(ii) how to obtain the source patch*.

For evaluating the two questions, we employ a Grad-CAM [45] module to locate the salient region in the image, and perform a series of studies about the saliency information for mixing. As a consequence, for (i), we find that the saliency-guided location surpasses that in CutMix, which keeps the location consistent in the two mixing images; while randomly determining the pasting location further surpasses the saliency-guided location. This indicates that the saliency information indeed facilitates the pasting location determining, but is defeated by the random location in terms of the data diversity. For (ii), the cropped patch

from the salient region only achieves similar performance with the randomly cropped patch. How to obtain a better image patch for mixing still remains an unsolved question. As shown in Fig. 1(a), we deduce the cutting manner for obtaining the image patch is easy to cause *label misallocation* due to the semantic inconsistency between the cropped patch and the whole source image, and *object information missing* which is verified in our experiment. Based on the above clues, we propose a novel and effective data mixing method, namely ResizeMix, which directly resizes the image and pastes the resized patch on another image. ResizeMix eliminates the label misallocation issue and preserves substantial information for mixing. The proposed ResizeMix consistently outperforms CutMix [51] and latter saliency-guided methods [22, 29, 50] on both CIFAR and ImageNet classification tasks. When transferred to the MS-COCO object detection task, the model trained on ImageNet with ResizeMix shows evident advantages over CutMix.

We summarize our contributions as follows.

1. Considering saliency information is widely used in recent mixing-based augmentation methods, we systematically check the importance of the saliency information, and find that saliency information is not so necessary for mixing data.

2. We verify that cropping the patch for mixing is easy to cause label misallocation and object information missing, and propose a new mixing method ResizeMix, which resolves the two issues by directly resizing the image for mixing.

3. The proposed ResizeMix shows evident advantages over CutMix and the saliency-guided methods on both image classification and object detection tasks without any additional computation cost, which even outperforms most costly search-based automatic augmentation methods.

2. Related Work

Cutting- and Mixing- based Data Augmentation The goal of cutting augmentations is to make a network pay attention to the entire data like the dropout regularization [7, 18, 20, 46, 47]. Random erasing [55] selects a patch of an image and masks it out. The width and height of the patch need to be designed manually. Beyond this, Cutout [11] proposes to mask a region with a fixed-size square. Another type of augmentation methods are based on mixing data. Mixup [53] attempts to produce an element-wise convex combination of two images. Augmix [26] mixes up the images augmented by operations sampled from the spaces like AutoAugment [8] defined ones. Rather than mixing the element-wise convex, RICAP [49] randomly gets four patches from different images and combines them to a new sample. CutMix [51] randomly crops

a patch from one image and pastes it into the corresponding position of another image, which significantly improves the test accuracy and exceeds most augmentation methods on various datasets.

Saliency Guided Data Augmentation Recently, mixing-based augmentation methods are widely used to augment images because they do not require extra searching or training cost while bringing significant performance improvement of networks. For example, CutMix [51] significantly improves the test accuracy and exceeds the most automatic augmentation methods [8, 9, 33]. However, the cropping and pasting method may cause label misallocation when the cropped patch is from the background of the image. Some studies further improve the performance of CutMix by reserving patches with more saliency information when cropping and pasting the patch between two images. PuzzleMix [29], which proposes to optimize the position of the mixing mask and the saliency discounted transportation. SuperMix [10] uses the knowledge of a teacher to mix images on their salient regions. Somewhat differently, FMix [22] sets a threshold for the low-frequency parts in the image to get the saliency masks for mixing images. SaliencyMix [50] uses a saliency detection module to select the saliency source patch for mixing. However, they all need extra cost to find the saliency regions. Compared to these methods, we propose a convenient and effective approach that can preserve the object information of images.

Automated Data Augmentation Parallel with the success of neural architecture search [4, 15–17, 37, 57], automated augmentation methods start to develop rapidly. AutoAugment [8] attempts to search for better combinations of augmentation operations and their magnitudes. Due to its expensive search cost when implemented with reinforcement learning, PBA [27] with population evolution strategy and FastAA [33] with matching density are proposed to speed up training without reducing the performance. The augmentation combinations can be treated as a hyper-parameter optimization formulation. OHL-AA [34] tries to optimize the probability distribution of augmentations, while Faster AA [23] and DADA [32] use the differentiable optimization directly to search the combinations and magnitudes of augmentations, which can save lots of searching cost. Integrated with adversarial training [2, 21, 40], AdvAA [54] makes networks learn more hard data samples, in which the domain of dataset becomes more widespread. Different from the search or optimization strategies, RandAugment [9] reaches identical performance only set up two parameters with the same augmentation spaces. Overall, most of the automated augmentation methods need extra search or training cost to obtain better performance, while our proposed ResizeMix can promote the network performance without any additional cost.

Table 1. Checking results on CIFAR-100 with WideResNet-28-10 about different manners of obtaining and the locations to paste the source patch. The column of "Type" means how the source patch is obtained, cutting or resizing from the source image. The "Region" column indicates the region in the source image to generate the source patch. The "Pasting Region" column indicates the location in the target image to paste the source patch.

Row	Source Patch		Pasting Region	Top-1 Acc(%)
	Type	Region		
(1) Baseline	-	-	-	81.20
(2) CutMix [51]	Cut	Random	Corresponding	83.40
(3)	Cut	Random	Non-salient	83.93
(4)	Cut	Random	Salient	83.97
(5)	Cut	Random	Random	**84.14**
(6)	Cut	Non-salient	Random	83.93
(7)	Cut	Salient	Random	84.07
(8)	Cut	Random	Random	84.14
(9) ResizeMix	Resize	Whole	Random	**84.31**

3. Checking the Importance of Saliency Information for Mixing Data

In this section, we systematically check whether the saliency information is necessary for mixing data. First, we introduce the preliminaries for our checking process in Sec. 3.1. Then we check the importance of saliency information from two perspectives, *i.e.* where to paste the source patch in Sec. 3.2 and how to obtain the source patch in Sec. 3.3.

3.1. Preliminaries

We use $I_s \in \mathbb{R}^{W \times H}$ and $I_t \in \mathbb{R}^{W \times H}$ to denote the source and target image respectively. We denote the source patch obtained from the source image as $P \in \mathbb{R}^{W_P \times H_P}$, while the patch cropped from the salient region as P_s, from the non-salient region as P_{ns}, and from a random region as P_r.

We employ a Grad-CAM [45] module to obtain the salient and non-salient pixels in the image by calculating the heatmap. The Grad-CAM module is connected to the end of the backbone network. Specifically, C_s represents a set of salient pixel coordinates where the activation value of the heatmap is greater than a certain upper threshold t_u; on the contrary, C_{ns} represents a non-salient coordinate set where the activation value is under a lower threshold t_l. They are defined as

$$
\begin{aligned}
C_s &= \{(x,y)|A(x,y) \geq t_u\}, \\
C_{ns} &= \{(x,y)|A(x,y) \leq t_l\},
\end{aligned}
\tag{1}
$$

where (x,y) denotes the coordinate of a pixel in the image, and $A(x,y)$ denotes the activation value at the position of (x,y).

We use $R(x_l, x_r, y_b, y_t)$ to denote a region of the image, and x_l, x_r, y_b, y_t represent the left, right, bottom and top
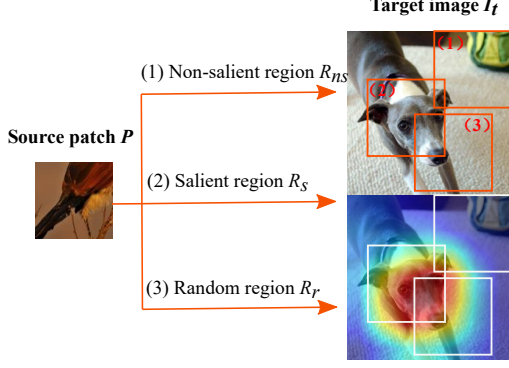
Figure 2. Three different regions to paste the source patch, including the non-salient, salient, and random region.

Figure 3. Three different manners of obtaining the source patch.

boundaries of the region. A salient region is denoted as R_s, whose geometric center is a salient pixel sampled from C_s. A non-salient region is denoted as R_{ns}, whose center is a non-salient pixel sampled from C_{ns}. And we use R_r to denote a random region whose center is randomly sampled from the whole image. The relationship of the geometric center (x_c, y_c) and the region boundaries is as follows,

$$x_c = \frac{x_l + x_r}{2}, \quad y_c = \frac{y_d + y_u}{2}. \quad (2)$$

We define the operation of pasting the source patch P to the region $R(x_l, x_r, y_b, y_t)$ in the target image I_t as $Paste(P, I_t, R)$,

$$Paste(P, I_t, R) : I_t[R] = I_t[x_l : x_r, y_b : y_t] = P. \quad (3)$$

CutMix [51] randomly crops a patch from the source image and pastes it to the target image. It can be formulated as:

$$\begin{aligned} P &= I_s[R_r], \\ I_m &= Paste(P, I_t, R_r), \\ l_m &= \lambda l_s + (1 - \lambda) l_t, \end{aligned} \quad (4)$$

where $R_r = (x_l, x_r, y_b, y_t)$ denotes a random region where to crop the source patch and to paste on the target image. The region R_r is the same in the source and target image. l_s, l_t and l_m denote the ground truth labels of the source, target and mixed image respectively. λ is computed as the ratio of the source patch and the target image, which is formulated as,

$$\lambda = \frac{W_P * H_P}{W * H}. \quad (5)$$

Since the random cropping manner proposed in CutMix may obtain the patch from the background of the image, which leads to label misallocation. To alleviate the shortcoming, some works [22, 29, 50] make use of the image saliency information to guide the mixing process. Considering the saliency information obtaining is usually complicated and costly, we systematically check the importance of the saliency information for mixing-based data augmentation from the following two aspects.
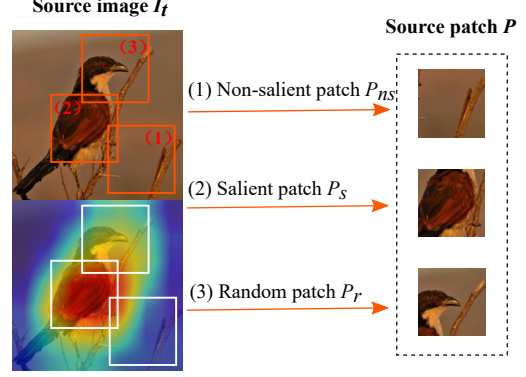
3.2. Checking Saliency Information for "Where to Paste the Source Patch"

To study the location for pasting the source patch, we crop the source patch from a random region of the source image as P_r, and paste it on various regions in the target image. Fig. 2 shows three different kinds of locations: (1) non-salient region R_{ns}, (2) salient region R_s, and (3) random region R_r.

The results of mixing data with three different patch pasting locations are shown in Row (3)-(5) of Tab. 1. We observe that the results of pasting the source patch to the non-salient region in Row (3) and the salient region in Row (4) both surpass the result of CutMix [51] in Row (2). The source patch is paste to the corresponding location of the target image in CutMix. The salient or non-salient region are both more diverse than the unique corresponding location, which leads to more various mixed images. It is notable that the settings of the salient and non-salient region show similar results. This indicates the network can always extract a part of information from the target image. Therefore, the saliency information for where to paste the source patch is not so necessary. Row (5) is the result of pasting the random patch to a random region in the target image, which further surpasses both the salient and non-salient region guided ones. This indicates the random region has more diversity for mixing the images, which contains both the salient and non-salient regions.

3.3. Checking Saliency Information for "How to Obtain the Source Patch"

In this section, we check whether saliency information is necessary for obtaining the source patch from the source image. As shown in Fig. 3, we crop patches from three different regions of the source image, i.e. the salient, non-salient and random region. The source patch is paste to a random region R_r of the target image.

Row (6)-(8) in Tab. 1 show the results of three different types of the source patch obtaining. We find that the result of the salient patch in Row (7) surpasses the non-salient
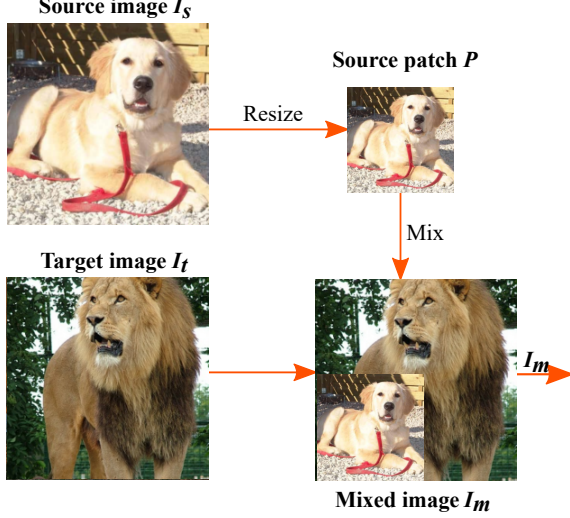
Figure 4. Process of ResizeMix. The source image is resized to a smaller patch and the patch is pasted to the target image, which generates the mixed image.

patch in Row (6). This is because the salient patch contains more information of the object corresponding to the allocated label than the non-salient patch. And if the non-salient patch contains too little information of the labeled object, this patch will lead to the problem of label misallocation. The salient patch is less possible to cause the misallocation. However, the result in Row (8) with the source patch cropped from a random region further outperforms the salient patch setting. This illustrates the random cropping manner can cover more regions with the labeled object preserved, while the salient patch only focuses on a smaller region; thus the random patch leads to more data diversity and achieves a better result. However, the random cropping strategy still carries the issue of label misallocation, how to better obtain the source patch remains an unsolved problem.

4. ResizeMix

Based on the checking results in Sec. 3, we observe that pasting the source patch in a random region of the target image leads to the best performance. For the cutting-based strategy of obtaining the source patch, when the cutting location covers more parts of a image, the label misallocation is aggravated as some patches contain no labeled object; when the cutting location focuses on the salient region to avoid label misallocation, the diversity of the mixed image decreases and some information of the source image is lost. The two issues of label misallocation and object information missing cannot be solved simultaneously under the cutting-based strategy.

To tackle the above two problems, we propose a new data mixing method ResizeMix. As shown in Fig. 4, we directly resize the whole source image to a smaller scale as

the source patch instead of cropping a patch from a local region. Then the source patch is pasted to a random region of the target image. ResizeMix can avoid the problem of label misallocation while the complete object information of the source image is preserved.

Specifically, we first resize the source image I_s to a smaller sized patch P by a scale rate of τ, which is defined as

$$P = T(I_s), \tag{6}$$

where $T()$ denotes the resizing operation and the scale rate τ is sampled from the uniform distribution $\tau \sim U(\alpha, \beta)$, where α and β denote the lower and upper bound of the range respectively. Then we paste the resized patch P into a random region R_r in the target image. This mixing operation introduces no additional computation cost, as the scale rate and the pasting region are both obtained randomly. The image mixing is formulated as

$$I_m = Paste(P, I_t, R_r). \tag{7}$$

We mix the source image label l_s and the target image label l_t according to the image mixing ratio λ,

$$l_m = \lambda l_s + (1 - \lambda)l_t, \tag{8}$$

where λ is defined by the size ratio of the patch and the target image, *i.e.* $\lambda = \frac{W_P * H_P}{W * H}$. W, H and W_P, H_P denote the width and height of the target image and the source patch respectively. As P is resized from the source image with the scale rate of τ, the relationship of W and W_P is $W_P = \tau * W$; the same as H and H_P. Therefore, λ and τ satisfy:

$$\lambda = \tau^2. \tag{9}$$

5. Experiments

In this section, we first study the effect of ResizeMix on image classification in Sec. 5.1. Then, we evaluate the generalization ability of the model pre-trained on ImageNet with ResizeMix by applying it on object detection in Sec. 5.2. Finally, we conduct some ablation studies and analysis in Sec. 5.3.

5.1. Evaluation on Image Classification

We evaluate the performance of ResizeMix on image classification dataset including CIFAR-10 [31], CIFAR-100 [31] and ImageNet [43].

5.1.1 Experiments on CIFAR-10

The CIFAR-10 dataset contains 60,000 color images of 32×32 size with 10 classes. There are 50,000 images for training and 10,000 images for validation. We implement ResizeMix on two neural netowrks, *i.e.* WideResNet-28-10 [52] and Shake-Shake (26 2x96d) [19]. We train the

Table 2. Top-1 test accuracy rate (%) on CIFAR-10 classification with WideResNet-28-10 [52] (WRS28-10) and Shake-Shake (26 2x96d) [19] (SS-2×96d). "ResizeMix+" denotes ResizeMix equipped with RandAugment [9]. "Cost" represents the additional computation cost introduced by searching or adjusting augmentation strategies, and † denotes the cost estimated according to the description in the original paper. "GHs": GPU Hours.

Method	Cost (GHs)	WRS28-10	SS-2×96d
Baseline	0	96.13	97.14
AA [8]	5000	97.32	98.00
Fast AA [33]	3.5	97.30	98.00
PBA [27]	5	97.42	97.97
OHL-AA [34]	83.4†	97.39	-
RA [9]	0	97.30	98.00
Faster AA [23]	0.23	97.40	98.00
DADA [32]	0.1	97.30	98.00
Cutout [11]	0	96.90	97.14
CutMix [51]	0	97.10	97.62
FMix [22]	6†	96.38	-
SaliencyMix [50]	6†	97.24	-
ResizeMix	0	97.60	97.93
ResizeMix+	6	**98.10**	**98.47**

WideResNet-28-10 network for 200 epochs with a batch size of 256 using the stochastic gradient descent (SGD) optimizer. We use the Nesterov momentum [13] of 0.9, and the weight decay of 5×10^{-4}. The initial learning rate is 0.1 and decays with the cosine annealing schedule [39]. When training the Shake-Shake (26 2x96d) network, we set the total epochs as 1,800 and the batch size as 256 using the SGD optimizer. The initial learning rate is 0.01 and the weight decay is 1×10^{-3}. We set the parameters of α, β for limiting the resizing scale ratios defined in Sec. 4 as 0.1 and 0.8, which are used for determining the range of the patch resizing scale.

The top-1 test accuracy comparisons are shown in Tab. 2. We compare the results of our method with CutMix [51], and some saliency-guided mixing augmentations [22, 29, 50], as well as some automated augmentation methods [8, 9, 23, 33]. Our proposed ResizeMix augmentation outperforms CutMix [51] by 0.5% and it even outperforms the automated augmentation method AutoAugment [8] by 0.28% with WideResNet-28-10. It is worth noting that ResizeMix does not introduce any additional computation cost, while most saliency-guided or automated augmentation methods take additional cost to promote the performance.

5.1.2 Experiments on CIFAR-100

The CIFAR-100 dataset has the same number of images as CIFAR-10 but it contains 100 classes. We apply our method ResizeMix on the WideResNet-28-10 and Shake-Shake (26 2x96d) network. We use the same settings

Table 3. Top-1 test accuracy rate (%) on CIFAR-100 classification with WideResNet-28-10 and Shake-Shake (26 2x96d).

Method	Cost (GHs)	WRS28-10	SS-2×96d
Baseline	-	81.20	82.95
AA [8]	5000	82.91	**85.72**
Fast AA [33]	3.5	82.70	85.40
PBA [27]	5	83.27	84.69
RA [9]	0	83.30	-
Faster AA [23]	0.23	82.20	84.40
DADA [32]	0.2	82.50	84.70
Cutout [11]	0	81.59	84.0
CutMix [51]	0	83.40	85.0
FMix [22]	6†	82.03	-
SaliencyMix [50]	6†	83.44	-
Puzzle Mix [29]	12†	84.05	-
ResizeMix	0	84.31	85.26
ResizeMix+	6	**85.23**	85.60

Table 4. Top-1 test accuracy rate (%) on ImageNet classification with ResNet-50 and ResNet-101 networks.

Method	Cost (GHs)	ResNet-50	ResNet-101
Baseline	-	76.31	78.13
AA [8]	15,000	77.63	-
FastAA [33]	450	77.60	-
OHL-AA [34]	625†	78.93	-
RA [9]	0	77.60	-
Faster AA [23]	2.3	76.50	-
DADA [32]	1.3	77.50	-
Cutmix [51]	0	78.60	79.83
SaliencyMix [50]	280†	78.74	79.91
Puzzle Mix [29]	576†	77.51	-
ResizeMix	0	**79.00**	**80.54**

and hyper-parameters as the CIFAR-10 dataset to train WideResNet-28-10 and Shake-Shake (26 2x96d). Tab. 3 shows the CIFAR-100 performance comparisons of our proposed ResizeMix with other cutting method [11], mixing method [29, 50, 51] and automated augmentations. We observe that ResizeMix outperforms CutMix [51] by 0.87%. Compared to the automated augmentations, it surpasses AutoAugment [8] by 1.40% and RandAugment [9] by 1.01%.

5.1.3 Experiments on ImageNet

ImageNet [43] is a challenging and widely used dataset for image classification. It contains 1.2 million training images and 50,000 validation images with 1,000 classes. The input image size is set as 224×224. We train our method with the networks of ResNet-50 and ResNet-101 [25] for 300 epochs. We set the batch size as 512, the initial learning rate as 0.5, and the weight decay as 4×10^{-5}. The learn-

Table 5. Generalization ability comparisons on object detection between ResizeMix and CutMix [51]. The experiments are performed on two frameworks of SSD [38] and Faster-RCNN [41] on both MS-COCO [35] and Pascal VOC [14] datasets.

Backbone	ImageNet-Cls Top-1 ACC(%)	MS-COCO Detection		Pascal VOC Detection	
		SSD mAP(%)	Faster-RCNN mAP(%)	SSD mAP(%)	Faster-RCNN mAP(%)
ResNet-50	76.1	25.1	38.1	75.6	81.0
Cutmix [51]	78.6	24.9	38.2	76.1	81.9
ResizeMix	**79.0**	**25.5**	**38.4**	**77.3**	**82.0**

ing rate decays with the cosine annealing schedule. The ImageNet results are shown in Tab. 4. With the ResNet-50 network, the performance of ResizeMix surpasses Cut-Mix [51] by 0.4% and Puzzle Mix [29] by 1.49%. It outperforms the automated ones, AutoAugment [8] by 1.37%, Faster AA [23] by 2.5%. It is worth noting that AutoAugment needs the additional computation cost of 15,000 GPU hours while ResizeMix does not introduce any additional cost. For ResNet-101, the performance of ResizeMix exceeds the performance of CutMix by 0.71%, which achieves the top-1 accuracy rate of 80.54%.

5.2. Evaluation on Object Detection

For evaluating the generalization ability of our method, we use the ResizeMix pre-trained ResNet-50 [25] model as the backbone network of two object detection frameworks, *i.e*. Faster RCNN [41] and SSD [38]. We perform the experiments on both MS-COCO [35] and Pascal VOC [14] datasets. All the experiments are based on the object detection toolkit MMDetection [5]. For SSD training, the input image is resized to 300×300. The batch size is set as 64 for two datasets. It takes 24 epochs in total. Both VOC2007 and VOC2012 `trainval` (VOC07+12) are used for training, and the models are evaluated on the VOC 2007 benchmark. For Faser-RCNN training, the image scale is set as $(1333, 800)$ for MS-COCO and $(1000, 600)$ for Pascal VOC. It takes 12 epochs in total for MS-COCO and 4 epochs for Pascal VOC respectively. For all the other training hyper-parameters, we just follow the default settings defined in MMDetection.

As shown in Tab. 5, our ResizeMix shows great generalization ability under several object detection evaluation settings. Especially on the lightweight framework SSD, ResizeMix shows notable mAP promotion over the baseline network, 0.4% mAP on MS-COCO and 1.7% mAP on Pascal VOC.

5.3. Ablation Study and Analysis

In this section, we perform a series of ablation studies and analysis about ResizeMix and other mixing-based augmentations. We first study the advantage of resizing over cutting on preserving the source image information in Sec. 5.3.1. Then we combine RandAugment [9] with ResizeMix and further promote the performance in Sec. 5.3.2.

Table 6. Comparisons of the effects between resizing and cropping on the half input resolution training. The shown results are all the top-1 accuracies (%) on the validation set. The "Train" and "Val" column indicate the strategies of obtaining half-resolution input images for training and validation respectively. "RandCrop" means randomly cropping a patch from the image and "Resize" means resizing the whole image to a smaller patch. "CenterCrop" means cropping a patch at the center of the testing image.

Row	Train	Val	CIFAR-10 WRS28-10	CIFAR-100 WRS28-10	ImageNet ResNet-50
Baseline	-	-	96.13	81.20	76.31
(1)	RandCrop	Resize	71.80	35.84	63.59
(2)	RandCrop	CenterCrop	90.10	66.70	58.58
(3)	Resize	Resize	**92.06**	**71.90**	**63.85**

Next, we explore several settings of resizing scale rates in Sec. 5.3.3. Finally, we analyze the differences between ResizeMix and other mixing-based augmentations in Sec. 5.3.4.

5.3.1 Cutting *vs*. Resizing on Information Preserving

We get the conclusion from Sec. 3.3 that cutting a patch from the source image may cause the problem of object information missing. To further verify the different effects of cutting and resizing on data mixing, we implement the comparison experiments under half input resolution settings. Specifically, during training, the input image is processed to a half-resolution one by randomly cropping a patch from the image or resizing the image to a half size. The images for validation are processed to the half sizes as well. The half-resolution experiments aim at comparing the information preserving abilities between cutting and resizing.

As shown in Tab. 6, processing the training images into the half-resolution ones by resizing shows evident advantages over cutting. When the training images are processed by cutting, no matter the testing images are processed by resizing or cutting at the image center, the final performance cannot surpass that with resizing the training images. This further demonstrates that for obtaining a patch from the image, the manner of resizing preserves more effective information than cutting.

5.3.2 Effect of RandAugment on ResizeMix

We are the first to study the effect of automated data augmentation on mixing data augmentation by combining Re-

7

Table 7. The results of different RandAugment placing positions on CIFAR-100 with WideResNet-28-10. "Before" means placing RandAugment operations before ResizeMix, and "After" means placing RandAugment after ResizeMix.

Position	Baseline	ResizeMix	Before	After
Top-1(%)	81.2	84.31	83.47	**84.59**

sizeMix with RandAugment. To verify the impact of the position relationship between ResizeMix and RandAugment on the training performance, we place the RandAugment operations before and after ResizeMixrespectively. We perform the experiment on the CIFAR-100 dataset with WideResNet-28-10, and all the hyperparameter settings are the same as that in Sec. 5.1.2. As shown in Tab. 7, the performance of putting RandAugment before ResizeMix is worse than using ResizeMix individually. This indicates that performing RandAugment on two images independently before mixing leads the two images to different patterns, which destroys the naturality of the mixed image and hinders the network learning. While RandAugment is performed after the images are mixed, the performance of ResizeMix obtains further improvement. It can be concluded that adding RandAugment after ResizeMix is a stronger augmentation pipeline to obtain better performance. It is worth noting that both ResizeMix and RandAugment do not introduce any additional computation cost.

When equipped with RandAugment [9] and the batch augmentation strategy [28, 34, 54] (the enlarging scale is set as 2 in our experiments), ResizeMix+ achieves top-1 accuracy rates of 98.10% with WideResNet-28-10 and 98.47% with Shake-Shake (26 2x96d) on CIFAR-10 in Tab. 2. And ResizeMix+ also achieves the new state-of-the-art performance of **85.23%** on CIFAR-100 with WideResNet-28-10 in Tab. 3.

Table 8. Comparison with different resizing scale ranges on CIFAR-100 with WideResnet-28-10.

Range	Baseline	0.1-0.9	0.1-0.8	0.1-0.7	0.2-0.8
Top-1(%)	81.20	83.91	**84.31**	83.72	83.70

5.3.3 Studying Resizing Scales

In this section, we study the settings of the resizing scale ratio. Since the scale ratio τ is randomly sampled from the uniform distribution $U(\alpha, \beta)$, we set different α and β to limit the range of ratio τ. Tab. 8 shows the results of different α and β settings. All the experiments are performed on CIFAR-100 with WideResNet-28-10, and all the settings are the same as that in Sec. 5.1.2. We observe that when setting α as 0.1 and β as 0.8 obtains the best performance, which is adopted to all the experiments with ResizeMix.
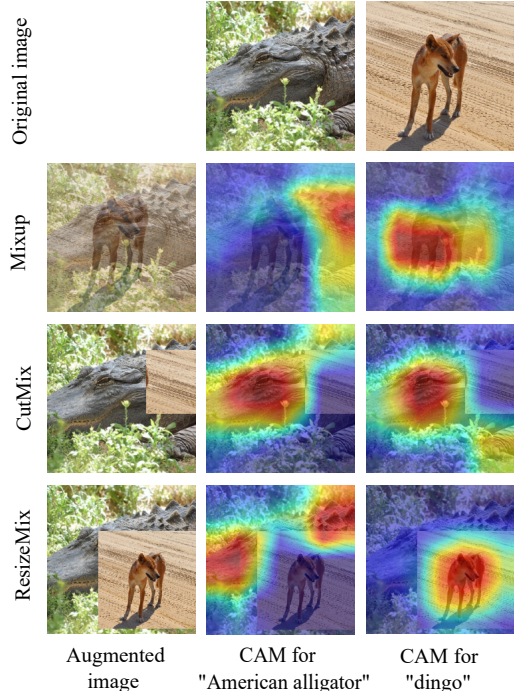

Figure 5. CAM visualization on "American alligator" and "dingo" using different augmentations.

5.3.4 Analysis on Different Mixing Methods

We visualize the CAM [56] heatmaps of images mixed with different methods. As shown in Fig. 5, the first row are the original images, and the first column on the left are the mixed images of various mixing methods including Mixup [53], CutMix [51], and ResizeMix. And the next two columns show the CAM heatmaps of categories "American alligator" and "dingo" respectively.

We observe that though the Mixup-generated image contains the informations of both categories, the mixed image in unnatural campared with real-life images. CutMix pastes a random patch of the source image into another image, but the patch is more likely to contain no information of "dingo", which leads to the problem of label misallocation. The network cannot locate the region corresponding to the label "dingo" and this will mislead the network learning. However, ResizeMix obtains the patch preserving all the information of the source image "dingo", which effectively eliminates label misallocation.

6. Conclusion

In this paper, we systematically study the CutMix-based data augmentation methods, and find that the saliency information of mixing data is not so necessary. Moreover, we conclude that the cutting-based data mixing strategies cannot avoid label misallocation and object information missing simultaneously. To tackle the two intractable problems, we propose an effective method, namely ResizeMix, which

directly resizes the image to a small patch and mixes it with another image. The proposed method shows evident advantages over previous methods on various image classification and object detection benchmarks.

Acknowledgement

References

[1] Dan Alistarh, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: randomized quantization for communication-optimal stochastic gradient descent. *arXiv:1610.02132*, 2016. 1

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv:1711.04340*, 2017. 3

[3] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *JMLR*, 2012. 2

[4] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 3

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019. 7

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 1

[7] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2019. 2

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *CVPR*, 2018. 1, 2, 3, 6, 7

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020. 3, 6, 7, 8

[10] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M Nasrabadi. Supermix: Supervising the mixing data augmentation. *arXiv:2003.05034*, 2020. 3

[11] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 2017. 2, 6

[12] Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *IJCAI*, 2015. 2

[13] Timothy Dozat. Incorporating nesterov momentum into adam. 2016. 6

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 7

[15] Jiemin Fang, Yuzhu Sun, Kangjian Peng, Qian Zhang, Yuan Li, Wenyu Liu, and Xinggang Wang. Fast neural network adaptation via parameter remapping and architecture search. In *ICLR*, 2020. 3

[16] Jiemin Fang, Yuzhu Sun, Qian Zhang, Yuan Li, Wenyu Liu, and Xinggang Wang. Densely connected search space for more flexible neural architecture search. In *CVPR*, 2020. 3

[17] Jiemin Fang, Yuzhu Sun, Qian Zhang, Kangjian Peng, Yuan Li, Wenyu Liu, and Xinggang Wang. Fna++: Fast network adaptation via parameter remapping and architecture search. *TPAMI*, 2020. 3

[18] RCNN Faster. Towards real-time object detection with region proposal networks. *NeurIPS*, 2015. 2

[19] Xavier Gastaldi. Shake-shake regularization. *arXiv:1705.07485*, 2017. 5, 6

[20] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *NeurIPS*, 2018. 2

[21] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. Deligan: Generative adversarial networks for diverse and limited data. In *CVPR*, 2017. 3

[22] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, and Adam Prügel-Bennett Jonathon Hare. Fmix: Enhancing mixed sample data augmentation. *arXiv:2002.12047*, 2020. 2, 3, 4, 6

[23] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. *arXiv:1911.06987*, 2019. 3, 6, 7

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[26] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *ICLR*, 2020. 2

[27] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *ICML*, 2019. 3, 6

[28] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: better training with larger batches. *CVPR*, 2020. 8

[29] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. *ICML*, 2020. 2, 3, 4, 6, 7

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[32] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxing

Yang. Dada: Differentiable automatic data augmentation. *arXiv:2003.03780*, 2020. 3, 6

[33] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *NeurIPS*, 2019. 3, 6

[34] Chen Lin, Minghao Guo, Chuming Li, Xin Yuan, Wei Wu, Junjie Yan, Dahua Lin, and Wanli Ouyang. Online hyper-parameter learning for auto-augmentation strategy. In *ICCV*, 2019. 3, 6, 8

[35] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 7

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1

[37] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 3

[38] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 7

[39] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017. 2, 6

[40] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *CVPR*, 2018. 3

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 7

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *LNCS*, 2015. 1

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5, 6

[44] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1

[45] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 2, 3, 10

[46] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 2

[47] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014. 2

[48] Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *WACV*, 2019. 2

[49] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. In *ACML*, 2018. 2

[50] AFM Uddin, Mst Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. *arXiv:2006.01791*, 2020. 2, 3, 4, 6

[51] 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 *ICCV*, 2019. 1, 2, 3, 4, 6, 7, 8

[52] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv:1605.07146*, 2016. 5, 6

[53] H Zhang, M Cisse, Y Dauphin, and D Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2018. 2, 8

[54] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. *ICLR*, 2020. 3, 8

[55] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 2

[56] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 8

[57] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018. 3

A. Appendix

A.1. Details of Obtaining Salient and Non-salient Regions

We first obtain the heatmap of the input image by using the Grad-CAM [45] module. We denote the activation values of the heatmap as A. Two thresholds t_u and t_l are set as the maximum and minimum activation values of A,

$$t_u = max(A), \quad t_l = min(A). \tag{10}$$

As there are many pixels which hold the activation values of the maximum or minimum values, we get the sets of salient and non-salient pixel coordinates C_s and C_{ns} as

$$
\begin{aligned}
C_s &= \{(x,y)|A(x,y) \geq t_u\}, \\
C_{ns} &= \{(x,y)|A(x,y) \leq t_l\},
\end{aligned}
\tag{11}
$$

where $A(x,y)$ denotes the activation value of the heatmap at the coordinate of (x,y).

We obtain the salient region $W_P \times H_P$ of a image as follows. We first randomly sample a coordinate (x_c, y_c) as the geometry center of the region from C_s, *i.e.*, $(x_c, y_c) \in C_s$. Then we calculate the boundaries of R_s as

$$
\begin{aligned}
x_l &= \lceil x_c - \frac{W_P}{2} \rceil, & x_r &= \lfloor x_c + \frac{W_P}{2} \rfloor, \\
y_b &= \lceil y_c - \frac{H_P}{2} \rceil, & y_t &= \lfloor y_c + \frac{H_P}{2} \rfloor,
\end{aligned}
\tag{12}
$$

where x_l, x_r, y_b, y_t denote the left, right, bottom and top boundary of the salient region R_s. Finally, we adjust these boundaries to guarantee the whole region is within the im-
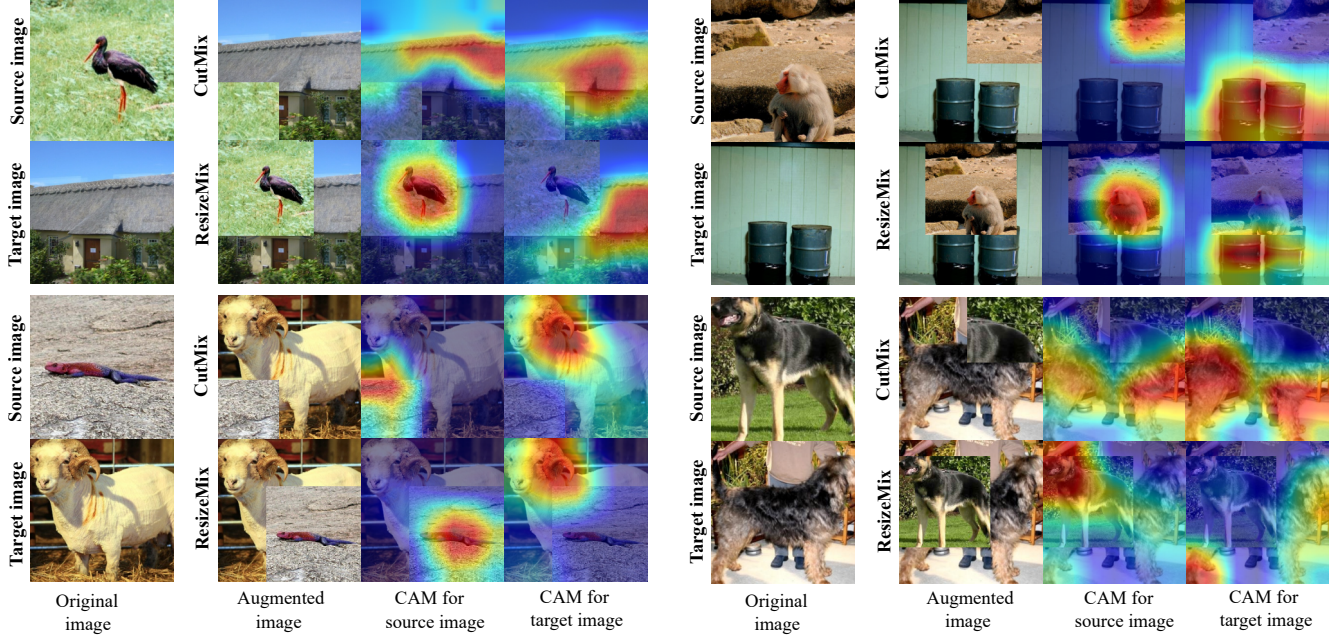
Figure 6. More visualization comparisons between CutMix and ResizeMix.

age. For x_l and x_r,

$$
\begin{aligned}
if \ x_l \le 0, & \begin{cases} x_l & = 0, \\ x_r & = W_P, \end{cases} \\
if \ x_r \ge W, & \begin{cases} x_l & = W - W_P, \\ x_r & = W; \end{cases}
\end{aligned}
\tag{13}
$$

For y_b and y_t,

$$
\begin{aligned}
if \ y_b \le 0, & \begin{cases} y_b & = 0, \\ y_t & = H_P, \end{cases} \\
if \ y_t \ge H, & \begin{cases} y_b & = H - H_P, \\ y_t & = H, \end{cases}
\end{aligned}
\tag{14}
$$

where W and H denote the width and height of the target image. The non-salient region R_{ns} can be obtained in the same way.