ResizeMix: Mixing Data with Preserved Object Information and True Labels

Jie Qin\textsuperscript{1,2}, Jiemin Fang\textsuperscript{3,4,*}, Qian Zhang\textsuperscript{5}, Wenyu Liu\textsuperscript{4}, Xingang Wang\textsuperscript{2†}, Xinggang Wang\textsuperscript{4}

\textsuperscript{1}School of Artificial Intelligence, University of Chinese Academy of Sciences
\textsuperscript{2}Institute of Automation, Chinese Academy of Sciences
\textsuperscript{3}Institute of Artificial Intelligence, Huazhong University of Science and Technology
\textsuperscript{4}School of EIC, Huazhong University of Science and Technology
\textsuperscript{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 \textit{label misallocation} and \textit{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. \textit{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-

\textsuperscript{*}Equal contribution. The work was performed during the internship of J. Qin and J. Fang at Horizon Robotics.
\textsuperscript{†}Corresponding author.
to obtain the source patch 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 hyperparameter 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], AdvaAA [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

| Row | Source Patch Type | Region | Pasting Region | Top-1 Acc(%) |
|-----|-------------------|--------|----------------|-------------|
| (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

$$C_s = \{(x, y) | A(x, y) \geq t_u\},$$

$$C_{ns} = \{(x, y) | A(x, y) \leq t_l\},$$

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
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 pasted 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
Specifically, we first resize the source image \( I_s \) to a smaller sized patch \( P \) by a scale rate of \( \tau \), which is defined as

\[
P = T(I_s),
\]

where \( T() \) denotes the resizing operation and the scale rate \( \tau \) is sampled from the uniform distribution \( \tau \sim U(\alpha, \beta) \), where \( \alpha \) and \( \beta \) 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

\[
I_m = \text{Paste}(P, I_t, R_r).
\]

We mix the source image label \( l_s \) and the target image label \( l_t \) according to the image mixing ratio \( \lambda \),

\[
l_m = \lambda l_s + (1 - \lambda)l_t,
\]

where \( \lambda \) is defined by the size ratio of the patch and the target image, i.e. \( \lambda = \frac{W_P}{W_s} \), \( W_s \), \( H_s \) 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 \( \tau \), the relationship of \( W \) and \( W_P \) is \( W_P = \tau \ast W \); the same as \( H \) and \( H_P \). Therefore, \( \lambda \) and \( \tau \) satisfy:

\[
\lambda = \tau^2.
\]

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 \( 32 \times 32 \) size with 10 classes. There are 50,000 images for training and 10,000 images for validation. We implement ResizeMix on two neural networks, i.e. WideResNet-28-10 [52] and Shake-Shake (26 x296d) [19]. We train the

![Diagram](image-url)
Table 2. Top-1 test accuracy rate (%) on CIFAR-10 classification with WideResNet-28-10 [32] (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 $5 \times 10^{-4}$. 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 $1 \times 10^{-3}$. We set the parameters of $\alpha$, $\beta$ 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 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 $4 \times 10^{-5}$. The learn-
sizeMix and further promote the performance in Sec. 5.3.2. Then we combine RandAugment [9] with Re-
sizing ability under several object detection evaluation set-
tings. Especially on the lightweight framework SSD, Re-
sizeMix shows notable mAP promotion over the baseline
network, 0.4% mAP on MS-COCO and 1.7% mAP on Pas-
scraping the source image information in
augmentations. We first study the advantage of resizing
over cutting on preserving the source image information.

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 SSD mAP(%) | Faster-RCNN mAP(%) | Pascal VOC Detection 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 |

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 re-
sizing 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 aug-
mentation on mixing data augmentation by combining Re-
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 ResizeMix respectively. 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(%)  | 81.2     | 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 $\tau$ is randomly sampled from the uniform distribution $U(\alpha, \beta)$, we set different $\alpha$ and $\beta$ to limit the range of ratio $\tau$. Tab. 8 shows the results of different $\alpha$ and $\beta$ 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 $\alpha$ as 0.1 and $\beta$ as 0.8 obtains the best performance, which is adopted to all the experiments with ResizeMix.

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 is unnatural compared 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

We thank Liangchen Song for the discussion and assistance.

References

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

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

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

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

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

[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.

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

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

[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.

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

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

[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.

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

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

[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.

[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.

[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.

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

[19] Xavier Gathaldi. Shake-shake regularization. arXiv:1705.07485, 2017.

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

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

[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.

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

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

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

[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.

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

[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.

[29] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. IJCAI, 2020.

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

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

[32] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxing Wu, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark. arXiv:1906.07155, 2019.
As there are many pixels which hold the activation values of the heatmap as 
the geometry center of the region from boundaries to guarantee the whole region is within the im-

\[ x, y \] denote the left, right, bottom and top boundaries of \( R \). Then we calculate the boundaries of \( W \) as

\[ s = t = (r, t) = (u, s) \]

\[ c \]

\[ A(x, y) \] denotes the activation value of the heatmap at the coordinate of \( (x, y) \).

\[ C_r = \{ (x, y) | A(x, y) \leq t_r \} \]

\[ C_s = \{ (x, y) | A(x, y) \geq t_u \} \]

\[ C_{ns} = \{ (x, y) | A(x, y) \leq t_l \} \]

\[ C_s = \{ (x, y) | A(x, y) \geq t_u \} \]

\[ C_{ns} = \{ (x, y) | A(x, y) \leq t_l \} \]

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

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

\[ C_s = \{ (x, y) | A(x, y) \geq t_u \} \]

\[ C_{ns} = \{ (x, y) | A(x, y) \leq t_l \} \]

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 \( (c_x, c_y) \) 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

\[ x_l = \left\lfloor x_c - \frac{W_p}{2} \right\rfloor \]

\[ y_l = \left\lfloor y_c - \frac{H_p}{2} \right\rfloor \]

\[ x_r = \left\lceil x_c + \frac{W_p}{2} \right\rceil \]

\[ y_r = \left\lceil y_c + \frac{H_p}{2} \right\rceil \]

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 $x_r$,

$$
\begin{align*}
if \ x_l \leq 0, & \quad x_l = 0, \\
& \quad x_r = W_P, \\
if \ x_r \geq W, & \quad x_l = W - W_P, \\
& \quad x_r = W; \\
\end{align*}
$$

(13)

For $y_b$ and $y_t$,

$$
\begin{align*}
if \ y_b \leq 0, & \quad y_b = 0, \\
& \quad y_t = H_P, \\
if \ y_t \geq H, & \quad y_b = H - H_P, \\
& \quad y_t = H, \\
\end{align*}
$$

(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.