Cut-Thumbnail: A Novel Data Augmentation for Convolutional Neural Network

Tianshu Xie*
tianshuxie@std.uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, China

Xuan Cheng*
cs_xuancheng@std.uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, China

Xiaomin Wang†
xmwang@uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, China

Minghui Liu
minghuiliuuestc@163.com
University of Electronic Science and Technology of China
Chengdu, China

Jiali Deng
julia_d@163.com
University of Electronic Science and Technology of China
Chengdu, China

Tao Zhou
zhou_tao@uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, China

Ming Liu
csmliu@uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, China

ABSTRACT

In this paper, we propose a novel data augmentation strategy named Cut-Thumbnail, that aims to improve the shape bias of the network. We reduce an image to a certain size and replace the random region of the original image with the reduced image. The generated image not only retains most of the original image information but also has global information in the reduced image. We call the reduced image as thumbnail. Furthermore, we find that the idea of thumbnail can be perfectly integrated with Mixed Sample Data Augmentation, so we put one image's thumbnail on another image while the ground truth labels are also mixed, making great achievements on various computer vision tasks. Extensive experiments show that Cut-Thumbnail works better than state-of-the-art augmentation strategies across classification, fine-grained image classification, and object detection. On ImageNet classification, ResNet-50 architecture with our method achieves 79.21% accuracy, which is more than 2.8% improvement on the baseline.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; **Supervised learning by classification**;

KEYWORDS

classification, convolutional neural network, thumbnail, data augmentation, regularization

ACM Reference Format:
Tianshu Xie, Xuan Cheng, Xiaomin Wang, Minghui Liu, Jiali Deng, Tao Zhou, and Ming Liu. 2021. Cut-Thumbnail: A Novel Data Augmentation for Convolutional Neural Network. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475302

1 INTRODUCTION

In recent years, the deep convolutional neural network (CNN) has made remarkable achievements in computer vision, including image classification [15, 20, 28, 31], object detection [12, 14, 27], and semantic segmentation [3, 24]. However, its huge structure and massive parameters pose a challenge to the training of the network. Many data augmentation and regularization approaches have been proposed to solve this problem.

As an important technology to generate more useful data from existing ones, data augmentation can significantly enhance network performance. The most commonly used data augmentation methods are spatial transformations, including random scale, crop, flip and random rotation [20]. Cutout [6] and Random Erasing [41] randomly set black blocks or place noises in one or more areas. Color distortion [31] changes the brightness of training images. Mixup [40] and CutMix [38] combines the two images with different strategies, and two images' labels are also mixed. In a word, the most existing data augmentation techniques improve the generalization ability and robustness of the network by changing spatial or color information, adding noise, or mixing information from different images.

---

*Both authors contributed equally to this research.
†Xiaomin Wang is the corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475302

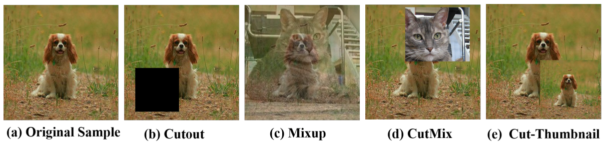


Figure 1: Comparison between existing data augmentation methods with Cut-Thumbnail.

In this paper, we introduce the idea of thumbnail into data augmentation and propose a novel augmentation strategy named Cut-Thumbnail. Figure 1 illustrates the comparison between existing data augmentation methods and Cut-Thumbnail. We reduce an image to a small size thumbnail and replace the original image's random area with it. Though being reduced, the thumbnail still contains most semantic information of the original image. By doing this, we not only make the network learn the features of images with different sizes, but also strengthen the network's capture of shape information. Furthermore, Cut-Thumbnail can be perfectly integrated with Mixed Sample Data Augmentation (MSDA), which refers to the combination of data samples according to a certain strategy. It is because when using a thumbnail to replace another image's random area, the thumbnail can completely contain the global information of its original image without taking up much semantics of another image. Therefore, we use a thumbnail to replace another image's random region, and mix their labels with certain weights. Besides, we find introducing two or more thumbnails into another image can also improve the network's effect on specific datasets. We specify a series of strategies around Cut-Thumbnail that will be presented in Section 3.

To demonstrate Cut-Thumbnail's effectiveness, we conduct extensive experiments on various CNN architectures, datasets, and tasks. On ImageNet [28], Cut-Thumbnail can improve the accuracy of ResNet-50 [15] from 76.32% to 79.21%, more effective than state-of-the-art method CutMix, which accomplishes 78.40%. On CIFAR100 [19], applying Cut-Thumbnail to ResNet-56 and WideResNet-28-10 [39] has improved the classification accuracy by +3.07% and +2.45%, respectively. Furthermore, On the CUB-200-2011 [35] dataset for the fine-grained classification task, Cut-Thumbnail increases the accuracy of ResNet-50 from 85.31% to 87.76%. On the Pascal VOC [8] dataset for the object detection task, our method increases the mAP of RetinaNet [23] from 70.14% to 72.16%.

To sum up, this paper makes the following contributions:

- We propose Cut-Thumbnail, a simple but effective data augmentation strategy *first* introducing the idea of thumbnail to data augmentation, which aims to make the network better learn the shape information.
- We combine Cut-Thumbnail with MSDA, the generated images of Thumbail are natural and contain most semantics of the reduced image.
- We conduct extensive experiments on classification, fine-grained classification and object detection. Comparing with state-of-the-art data augmentation methods, the experimental results demonstrate that our method achieves the best performance.

2 MOTIVATION

Shape v.s Texture: Recent studies have shown that CNN is texture-biased, *i.e.* CNN relys more on local texture rather than global shape in decision-making [1, 2, 9, 10, 29]. CNN can classify texture images well, but it is not sensitive to the shape of objects. For example, CNN tends to classify an image with a cat shape filled with an elephant skin texture as an elephant instead of a cat [10]. [10, 29] denote that improving the shape bias of CNN can improve the accuracy and robustness. However, the shape information contained in image is scarce and vague, making the network difficult to capture effective shape information.

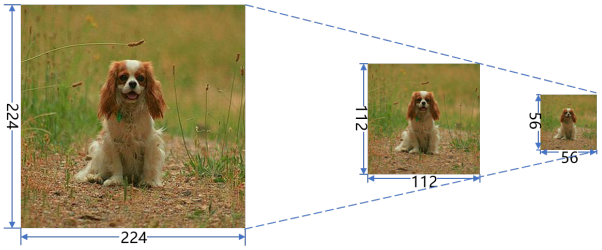
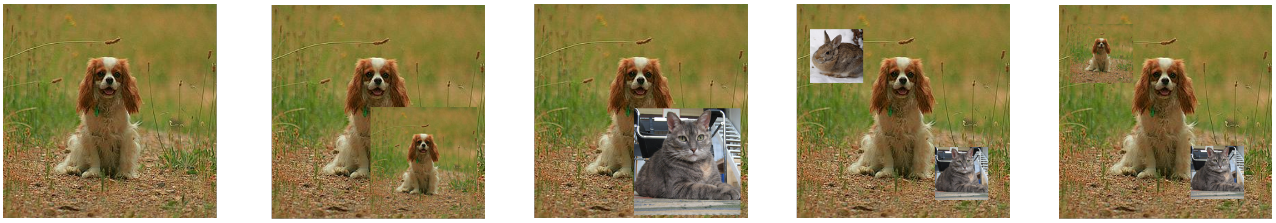


Figure 2: This image shows an example of reduced images that we call thumbnails. After reducing the image to a certain size 112×112 or 56×56, we can still recognize a dog in the image even though lots of local details are lost.

.

Different from [10] which consumes lots of computational overhead to generate images with little texture information, we push forward a novel data augmentation strategy to force the network to perceive more shape information by utilizing the property of thumbnail. As shown in Figure 2, we can still recognize that there is a dog in the image after reducing the image to a certain size. We call the reduced image as a thumbnail. Although lots of the texture details are lost, we can still identify the target in the thumbnail. This is because human usually relies on the global information of the image [7, 21], such as shape, for image classification, which is still reserved by the thumbnail. We consider using the shape information that the thumbnail contained is a simple but effective way to improve the shape bias of network. When we put the thumbnail on its original image, the network can learn both the shape information and the texture information simultaneously. To verify whether the network trained with Cut-Thumbnail can improve the shape bias of CNN, we conduct an experiment that can be seen in SubSection 3.4. The result shows that our network performs better on the dataset with the grayscale images that contain more shape information and less texture information than other methods.

Cut-Tumbnail v.s CutMix: While both replacing the image region, CutMix [38] tends to use a random patch from another image, which is quite possible to be uninformative. Cut-Tumbnail overcomes this problem by using the thumbnail, which retains most semantics of another image. Note that CutMix uses the patch of the image with no complete shape information, and just strengths the texture learning of CNN. While Cut-Thumbnail uses thumbnail that contains complete shape information and improves the shape bias of CNN. As can be seen in later experiments, this strategy



(a) Original Sample (b) Self Thumbnail (c) Mixed Single Thumbnail (d) Mixed Multiple Thumbnails (e) Other Strategy

Figure 3: Illustration of Cut-Thumbnail. We put a single thumbnail or multiple thumbnails on the thumbnail's original image or another image, and thus get different strategies.

significantly improves network performance and gets better results in classification, fine-grained classification, and object detection than CutMix.

3 OUR APPROACH

Cut-Thumbnail is an effective albeit simple data augmentation technique for CNN. For a given training sample (x, y) which $x \in \mathbb{R}^{W\times H\times C}$ denotes the training image and y denotes the training label, we get a thumbnail image $T(x)$ by simply taking one pixel out of a certain number of pixels of the image x. We put a single thumbnail or multiple thumbnails on the thumbnail's original image or another image, and thus get different strategies. Next, we will introduce them in turn.

3.1 Self Thumbnail (ST)

In our first strategy, we use the thumbnail $T(x) \in \mathbb{R}^{w\times h\times C}$ to replace a random region of the original image x and do not change the label, which is called Self Thumbnail as shown in Figure 3(b). For a given training sample (x_1, y_1), We define this operation as

$$\begin{aligned} \tilde{x} &= \mathrm{M} \odot x_1 + \Phi(T(x_1)) \\ \tilde{y} &= y_1 \end{aligned} \tag{1}$$

where $(\tilde{x}, \tilde{y})$ denote the generated sample, $\mathrm{M} \in \{0,1\}^{W\times H}$ is the binary mask indicating where to drop out and fill in from origin image and thumbnail, and $\odot$ is element-wise multiplication. To sample the binary mask M, we first sample the bounding box coordinates $\mathrm{B} = (r_x, r_y, r_w, r_h)$ indicating the cropping regions on x_1. The region B in x_1 is removed and filled in with the thumbnail $T(x_1)$. The box coordinates are uniformly sampled according to

$$\begin{aligned} r_x &\sim \mathrm{Unif}(0, W), r_w = w \\ r_y &\sim \mathrm{Unif}(0, H), r_h = h \end{aligned} \tag{2}$$

where w, h denote the width and height of the thumbnail $T(x_1)$, which are usually set to half the width and height of the original image. With the cropping region, the binary mask M is decided by filling with 0 within the bounding box B, otherwise 1. $\Phi(\cdot)$ denotes the padding operation that generates an image with the same size as x_1. Φ first generates a binary mask $\tilde{\mathrm{M}} = 1 - \mathrm{M}$, 1 is a binary mask filled with ones. The bounding box coordinates B still exists in $\tilde{\mathrm{M}}$, so we put the thumbnail $T(x_1)$ in B and the generated image is $\Phi(T(x_1))$. This strategy enables the network to learn the same image at different scales. In addition to the information obtained from the original image, the thumbnail can provide the global information for the training, which plays a guiding role in the network learning.

3.2 Mixed Single Thumbnail (MST)

The idea of thumbnail is very suitable for Mixed Sample Data Augmentation (MSDA), which involves combining data samples according to a certain policy to create an augmented data set. In our second strategy, one image's random region is replaced by another's thumbnail where their labels are multiplied by different weights and added, so that most of the generated images contain the information of two images as shown in Figure 3(c). We call this strategy as Mixed Single Thumbnail. For a pair of given training sample(x_1, y_1) and (x_2, y_2), we define this combining operation as

$$\begin{aligned} \tilde{x} &= \mathrm{M} \odot x_1 + \Phi(T(x_2)) \\ \tilde{y} &= (1-\lambda)y_1 + \lambda y_2 \end{aligned} \tag{3}$$

where M, Φ, and the size of thumbnail $T(x_2)$ is set in the same way as Self Thumbnail. This strategy combines the idea of thumbnail with MSDA, so that the network can learn the original information of one image and the complete information of another image simultaneously.

3.3 Mixed Multiple Thumbnails (MMT)

One of thumbnail's advantages is that it can introduce one image's complete semantics by occupying a small area of another image. Besides, unlike the simple overlay in Mixup, Cut-Thumbnail does not make the original image appear unnatural. Therefore, we can further expand Cut-Thumbnail's superiority by putting two or more thumbnails on another image as shown in Figure 3(d). Take adding n thumbnails to another image as an example, for given training samples (x_1, y_1) , (x_2, y_2) , ... , (x_n, y_n), we define this combining operation as

$$\begin{aligned} \tilde{x} &= \mathrm{M} \odot x_1 + \sum_{i=2}^{n} \Phi_i(T(x_i)) \\ \tilde{y} &= \lambda_1 y_1 + \sum_{i=2}^{n} \lambda_i y_i \end{aligned} \tag{4}$$

where M contains $n-1$ bounding box coordinates $\mathrm{B}_2, \mathrm{B}_3, \dots, \mathrm{B}_\mathrm{n}$ corresponding to $\Phi_2, \Phi_3, \dots, \Phi_n$. Unlike CutMix, the weight of image labels λ_i is not determined by the area of the thumbnails. It is because CutMix only adds the random parts of two images, but the thumbnail has most of the original image's semantics, it should have a higher weight, which can be seen in Section 4.5. Mixed Multiple Thumbnails can make each training image contain more images' global information, improving the network training efficiency. We find that Mixed Multiple Thumbnails significantly improves the network performance on datasets with the larger image size and less data volume like CUB-200-2011 [35].

During training, we randomly choose 80% of batches using Cut-Thumbnail rather than every batch. We call the rate of batches applying Cut-Thumbnail as *participation_rate*. Note that although our method has achieved a high level of improvement, we have not deliberately sought for the optimal combination of these strategies due to the limitation of time and computing resources. In other words, the potential of Cut-Thumbnail can be explored in future work.

3.4 Why Does Cut-Thumbnail Help?

Graysacle image recognition: The shape information contained in thumbnail, together with the texture bias of CNN, further motivates our proposed method that aims to improve the shape bias of CNN. To verify that Cut-Thumbnail can indeed make the network learn more shape information, we use grayscale images as the test set to compare the performance of the networks trained by different methods. Due to the lack of color information, graysacle images have less texture details, but its shape information is not affected. Therefore, the recognition of grayscale image requires the network to rely more shape information compared with color images.

Model	Greyscale Image(%)
ResNet-50(Baseline)	64.70
ResNet-50+CutMix	67.61(+2.91)
ResNet-50+ST	66.95(+2.25)
ResNet-50+MST	**68.63(+3.93)**

Table 1: Comparison of CutMix and Cut-Thumbnail on greyscale image with ResNet-50. ST denotes Self Thumbnail and MST denotes Mixed Single Thumbnail.

We transform all the images in ImageNet's verification set into grayscale images as a new verification set. The tested networks are trained on regular Imagenet training set with different methods, and the training details can be seen in SubSection 4.1. As demonstrated in Table 1, The network trained with Mixed Single Thumbnail(MST) has achieved the highest results(+3.93%) on grayscale image. The performance on grayscale image of our method shows that the shape information contained in the thumbnail is helpful for improving the shape bias of CNN. In the absence of texture information, the grayscale image recognition of network trained by MST is better than CutMix, which shows that our method can enhance the learning of image shape.

Network visualization: To analyze what the model trained with Cut-Thumbnail learns, we compute class activation mapping (CAM) for ResNet-50 model trained with ST and MST on ImageNet. We also show the CAM for models trained with baseline augmentation and CutMix for comparison in Figure 4. The models trained with ST and MST both tend to focus on large important regions, while the network trained by CutMix tends to focus on local regions. This proves that the network trained with Cut-Thumbnail is biased to the global information of the image for object recognition, which is different from CutMix that makes the network pay more attention to the local regions.

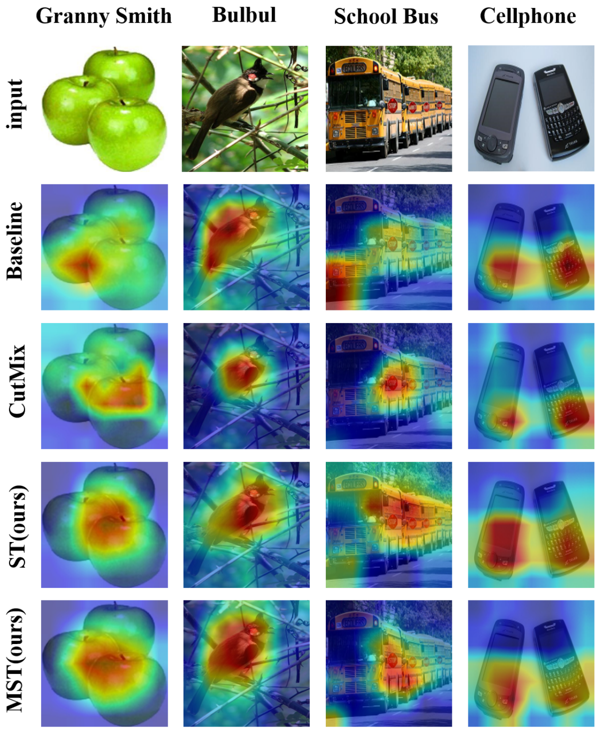


Figure 4: Class activation mapping (CAM) [42] for ResNet-50 model on ImageNet, with baseline augmentation, CutMix, ST or MST.

4 EXPERIMENT

In this section, we investigate the effectiveness of Cut-Thumbnail for several major computer vision tasks. We first conduct extensive experiments on image classification and fine-grained image classification. Next, we study the effect of Cut-Thumbnail on object detection. All experiments are performed with Pytorch [25] on Tesla M40 GPUs.

4.1 ImageNet Classification

ImageNet-1K [28] contains 1.2M training images and 50K validation images labeled with 1K categories. We use the standard augmentation setting for ImageNet dataset such as resizing, cropping, and flipping. For fair comparison, the model is trained from scratch for 300 epochs with batch size 256 and the learning rate is decayed by the factor of 0.1 at epochs 75, 150, 225, as done in CutMix [38]. We evaluate classification accuracy on the validation set and the highest validation accuracy is reported over the full training course following the common practice. For Self Thumbnail (ST) and Mixed Single Thumbnail (MST), we set the *thumbnail_size* to 112×112

Model	Method	Accuracy(%)
ResNet-18	baseline	69.90±0.09
	+CutMix	70.30±0.03
	+ST (ours)	**71.92±0.04**
	+MST (ours)	71.34±0.07
ResNet-50	baseline	76.32±0.02
	+Cutout	77.07±0.04
	+Mixup	77.42±0.06
	+AutoAugment*	77.63
	+DropBlock*	78.13±0.05
	+CutMix	78.40±0.04
	+ST (ours)	77.74±0.05
	+MST (ours)	**79.21±0.04**

Table 2: Summary of validation accuracy of the ImageNet classification results based on ResNet-18 and ResNet-50. ST denotes Self Thumbnail and MST denotes Mixed Single Thumbnail. We report average over 3 runs. '*' means results reported in the original paper.

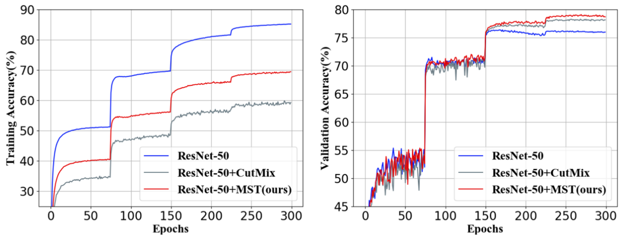
which is half of the image width and height, and λ in MST is set to 0.25. We explore the performance of different data augmentation methods on ResNet-18 and ResNet-50 [15]. The results are illustrated on Table 2.

Performance on ResNet-18: With ST, we improve the accuracy of ResNet-18 from 70.10% to 71.92% (+1.82%), which surpasses CutMix significantly. The improvement of CutMix on ResNet-18 is not obvious, we speculate that it is because the images generated by CutMix are relatively complex for ResNet-18 with weak learning ability. ST uses the image's own thumbnail to replace its random region without adding the extra image information, which may be more beneficial to the training of small networks like ResNet-18.

Performance on ResNet-50: ResNet-50 is a widely used CNN architecture for image recognition. We can observe that MST achieves the best result, 79.21% top-1 accuracy, among the considered augmentation strategies. Cutout [6] randomly masks square sections of the image. We set the mask size for Cutout to 112×112 and the location for dropping out is uniformly sampled. Inspired by Cutout, DropBlock [11] randomly drops some contiguous regions of a feature map. MST outperforms Cutout and DropBlock by +2.14% and +1.08%, respectively.

Mixup and CutMix are the successful variants of MSDA, which achieve excellent results in classification tasks. We set $\alpha = 1$ in both Mixup [40] and CutMix. MST outperforms Mixup and CutMix, by +1.79% and +0.81%, respectively. It shows that the images generated with thumbnail are more conducive to network learning.

Note that ST also achieves a great performance on ResNet-50. ST utilizes only one image's information, but it performs better than Mixup using multiple images' information. Besides, AutoAugment [5] uses reinforcement learning to find a combination of existing augmentation policies. ST, by simply putting the image's own thumbnail on itself, even exceeds the performance of AutoAugment, which demonstrates the effectiveness and generality of our method.



(a)Training accuracy on ImageNet (b)Validation accuracy on ImageNet

Figure 5: Training and validation accuracy comparison among baseline, CutMix and MST on ImageNet with ResNet-50.

Training and validation accuracy comparison among baseline, CutMix and MST on ImageNet with ResNet-50 can be seen in Figure 5. Due to the mixing of labels and images, the accuracy of MSDA methods such as Mixup and CutMix is far lower than the baseline, and MST is no exception. But the validation accuracy of CutMix and MST is much higher than baseline and the accuracy of MST is higher than that of CutMix, which shows that our method can significantly improve the generalization of the network.

4.2 Tiny ImageNet Classification

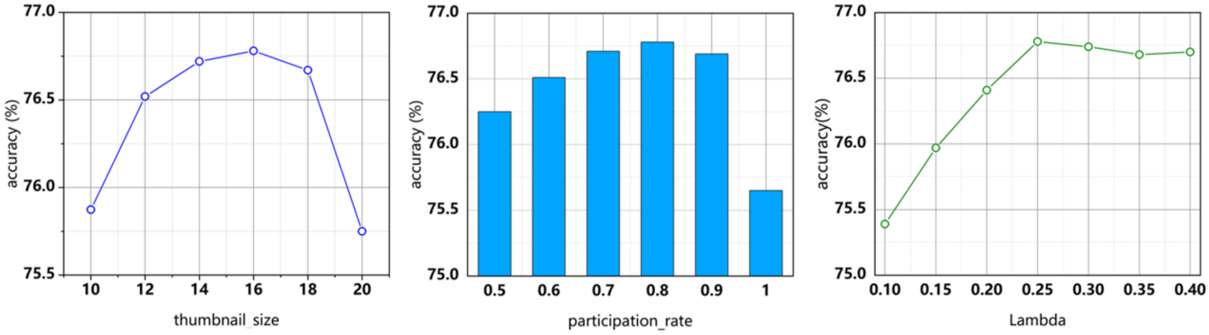
Tiny ImageNet dataset is a subset of the ImageNet dataset with 200 classes. Each class has 500 training images, 50 validation images, and 50 test images. All images are with 64×64 resolution. We test the performance of ResNet-110 on this dataset. The learning rate is initially set to 0.1 and decayed by the factor of 0.1 at epochs 150 and 225. For ST and MST, we set the size of the thumbnail to 32×32, and λ in MST is set to 0.25. The hole size of Cutout is set to 32×32. For Mixup and CutMix, the hyper-parameter α is set to 1.0. The results are summarized on Table 3. MST achieves the best performance 66.45% on Tiny ImageNet. This proves that our method is also generalized for datasets with different data sizes.

Model	Accuracy(%)
ResNet-110 (baseline)	62.42±0.02
ResNet-110+Cutout	64.71±0.09
ResNet-110+Mixup	65.34±0.14
ResNet-110+CutMix	66.13±0.02
ResNet-110+ST (ours)	64.85±0.04
ResNet-110+MST (ours)	**66.45±0.03**

Table 3: Comparison of state-of-the-art data augmentation methods on Tiny ImageNet with ResNet-110.

4.3 CIFAR Classification

The CIFAR10 [19] dataset collects 60,000 32×32 color images of 10 classes, each with 6000 images including 5,000 training images and 1,000 testing images. The CIFAR100 [19] dataset has the same number of images but 100 classes. To test the universality of our method



(a)thumbnail_size (b)participation_rate (c)λ

Figure 6: CIFAR100 validation accuracy against $particitpation\_rate$ $thumbnail\_size$, $particitpation\_rate$, and λ with ResNet-56. We set $thumbnail\_size = 16$, $particitpation\_rate = 0.8$, and $\lambda = 0.25$ as the default parameter settings.

under different network structures, ResNet-56 and WideResNet-28-10 are selected as baseline. For WideResNet-28-10, the learning rate is decayed by the factor of 0.1 at epochs 60, 120, 160; for ResNet-56, the learning rate is decayed by the factor of 0.1 at epochs 150, 225. For ST and MST, we set the $thumbnail\_size$ to 16×16, and λ in MST is set to 0.25. The hole size of Cutout is set to 16×16. For Mixup and CutMix, the hyper-parameter α is set to 1.0.

Model	Method	Accuracy(%)
ResNet-56	baseline	73.71±0.12
	+Cutout	74.64±0.15
	+Mixup	75.97±0.26
	+CutMix	76.57±0.13
	+ST (ours)	75.58±0.11
	+MST (ours)	**76.78±0.08**
WideResNet-28-10	baseline	81.06±0.03
	+Cutout	81.86±0.08
	+Mixup	82.57±0.12
	+CutMix	83.13±0.06
	+ST (ours)	81.41±0.04
	+MST (ours)	**83.35±0.05**

Table 4: Comparison of Top-1 accuracy of ResNet-56 and WideResNet-28-10 on the CIFAR100 validation set. MST obtains the best performance on both networks.

As shown in Table 4, on CIFAR100 dataset, MST provides better results over ResNet-56 and WideResNet-28-10 compared to Cutout, Mixup and CutMix. For ResNet-56, MST achieves a significant 3.07% improvement over the base model. ST outperforms Cutout on ResNet-56, showing that pasting the image's own thumbnail is better than black block. Generalization is an essential property of data augmentation methods, experiments show that our method is suitable for networks with different structures.

Model	Accuracy(%)
ResNet-56 (baseline)	94.00±0.14
ResNet-56+Cutout	94.80±0.18
ResNet-56+Mixup	95.01±0.16
ResNet-56+CutMix	**95.33±0.11**
ResNet-56+ST (ours)	95.03±0.09
ResNet-56+MST (ours)	95.24±0.12

Table 5: Impact of Cut-Thumbnail on CIFAR10 for ResNet-56.

As shown in Table 5, on CIFAR10 dataset, MST improves the performance by +1.24% on ResNet-56, but slightly lower than 1.33% of CutMix. We consider it may be that images in CIFAR10 are relatively simple with low pixels, the information provided by thumbnails is limited. But for CIFAR10 and CIFAR100 with images of very small sizes, our method can also achieve significant performance improvement, which demonstrates Cut-Thumbnail applies to various types of datasets.

4.4 Ablation Studies

We conduct ablation studies on CIFAR100 dataset using the same experimental settings of ResNet-56 in Subsection 4.3.

Analysis on $thumbnail\_size$: We evaluate Cut-Thumbnail with $thumbnail\_size \in \{10,12,14,16,18,20\}$. As shown in Figure 6(a), with the increasing of $thumbnail\_size$, the accuracy first rises and then decreases after reaching the highest when the $thumbnail\_size$ is 16, which is half of the image width or height. It denotes that the small thumbnail does not have enough semantics to guide network training, while the large one affects the semantics of the original image. Therefore, we generally select the thumbnail with the half width and height of the original image.

Effect of the $participation\_rate$: Specially, we call the ratio of batches using Cut-Thumbnail to all batches as the $participation\_rate$. As shown in Figure 6(b), when the participation_rate between 0.7 and 0.9, the difference in network performance is not obvious, but they are significantly better than the performance when the $participation\_rate$ is 1. This shows that our method is not sensitive to *participation_rate* as long as the *participation_rate* is higher than 0.7 and not to be 1. We consider the reason may be that the network needs to supplement a small number of normal images for comparative learning with thumbnail.

Exploration to λ: We test the effect of λ on the training, which is the weight multiplied by the image label. The results are given in Figure 6(c), when the λ is set to 0.25, the model performance is the best. Besides, the difference in network performance is not obvious either when the λ is between 0.25 and 0.35. We consider it because the thumbnail contains most semantics of the original image, so multiplying the thumbnail's label with a higher weight is beneficial to network training.

4.5 Fine-grained Image Classification

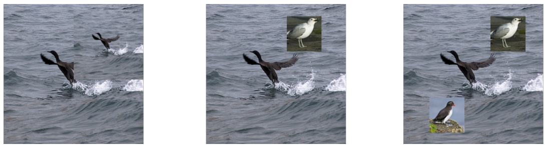
The fine-grained image classification aims to recognize similar subcategories of objects under the same basic-level category. The difference of fine-grained recognition compared with general category recognition is that fine-grained subcategories often share the same parts and usually can only be distinguished by the subtle differences in texture and color properties of these parts. CUB-200-2011 [35] is a widely-used fine-grained dataset which consists of images in 200 bird species. There are about 30 images for training for each class.

To verify the generalization of different types of computer vision tasks, we use ResNet-50 to test the performance of our method on CUB-200-2011. The training starts from the model pretrained on ImageNet. The mini-batch size is set to 16 and the number of training epoch is set to 95. The learning rate is initially set to 0.001 and decayed by the factor of 0.1 at epochs 30, 60 and 90. During network training, the input images are randomly cropped to 448×448 pixels after being resized to 600×600 pixels and randomly flipped. As shown in Table 6, with Mixed Double Thumbnail (MDT) which denotes two images' thumbnails are put on another image, we improve the accuracy of ResNet-50 from 85.31% to 86.72%(+1.41%), which surpasses previous data augmentation methods significantly. It also shows that the shape information has a high gain effect even on fine-grained image classification which requires more subtle differences.

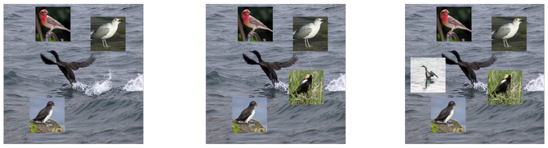
Model	Accuracy(%)
ResNet-50 (baseline)	85.31±0.21
ResNet-50+Cutout	85.68±0.13
ResNet-50+Mixup	85.91±0.31
ResNet-50+CutMix	86.12±0.16
ResNet-50+ST (ours)	85.72±0.11
ResNet-50+MST (ours)	86.56±0.16
ResNet-50+MDT (ours)	**86.72±0.12**

Table 6: Performance of data augmentation methods on CUB-200-2011. MDT denotes Mixed Double Thumbnial. Both MDT and MST outperform CutMix.

Performance of Mixed Multiple Thumbnails: Unlike the above classification datasets, the number of each category's images in CUB-200-2011 is smaller, and each image's resolution is higher. This drives us to put more than one thumbnail on another image, as shown in Figure 7. It is because putting more thumbnails on each image will not take up much semantics of the original image with such high resolution. We set the thumbnail size to 130×130, the weight of the origin image label λ_1 is set to 0.6, and the weight of the thumbnail label λ_i is set to 0.2. As shown in Figure 8, with the increase in the number of thumbnails, the accuracy first rises and then falls and achieves the highest (86.62%) when the number is two. Besides, we note that the original image still retains more than 60% of area when mixed with 5 thumbnails, so the network performance can still be significantly improved.



(a) Self Thumbnail (b) Mixed Single Thumbnail (c) Mixed Double Thumbnails



(d) Mixed Triple Thumbnails (e) Mixed Quadruple Thumbnails (f) Mixed Quintuple Thumbnails

Figure 7: Examples of training images using Mixed Multiple Thumbnails on CUB-200-2011.

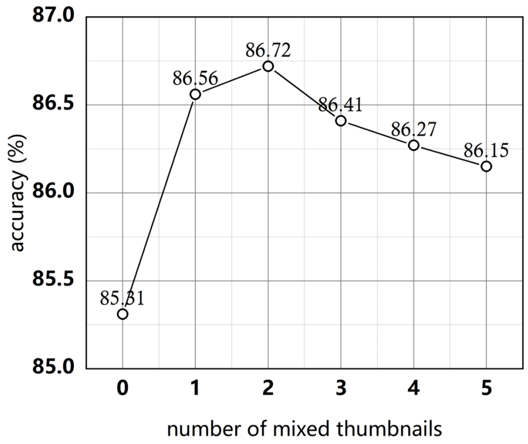


Figure 8: Performance of Mixed Multiple Thumbnails with different number of mixed thumbnails on CUB-200-2011. The network achieves the best performance when 2 thumbnails are pasted on each training image.

Transfer Learning of Pretrained Model: ImageNet pre-training is de-facto standard practice for many visual recognition tasks. We examine whether Cut-Thumbnail pre-trained models lead to better performances in fine-grained image classification. As shown in Table 7, the ResNet-50 with MST pre-trained model performs better than baseline. Furthermore, on the basis of taking the network pretrained by MST as the backbone, we have superimposed MST in the training, and the result achieves 87.76%, which is more than 2.45% improvement on the baseline.

Model	Accuracy(%)
ResNet-50 (baseline)	85.31±0.21
ResNet-50+MST	86.56±0.16
ResNet-50+MST pre-trained	86.17±0.12
ResNet-50+MST pre-trained+MST	**87.76±0.13**

Table 7: Fine-grained image classification results on CUB-200-2011 with different backbone models.

4.6 Object Detection in PASCAL VOC

In this subsection, we show Cut-Thumbnail can also be applied for training object detector in Pascal VOC [8] dataset. We use RetinaNet [23] framework composed of a backbone network and two task-specific subnetworks for the experiments. The ResNet-50 backbone which is responsible for computing a convolutional feature map over an entire input image is initialized with ImageNet-pretrained model and then fine-tuned on Pascal VOC 2007 and 2012 trainval data. Models are evaluated on VOC 2007 test data using the mAP metric. We follow the fine-tuning strategy of the original method.

Model	mAP(%)
RetinaNet (baseline)	70.14±0.17
RetinaNet+CutMix pre-trained	71.01±0.21
RetinaNet+ST pre-trained (ours)	71.01±0.15
RetinaNet+MST pre-trained (ours)	**72.16±0.19**

Table 8: Object detection results on Pascal VOC with RetinaNet. The model pre-trained with MST achieves the best accuracy.

As shown in Table 8, the model pre-trained with MST achieves the best accuracy (72.16%), +2.02% higher than the baseline performance(70.14%). It proves that our method is suitable for object detection task. Besides, CutMix works better than ST in Imagenet classification task, but the model pre-trained by ST performs equal to CutMix in object detection task. The results suggest that the model trained with Cut-Thumbnail can better capture the target objects.

5 RELATED WORK

Regularization: The regularization methods are effective for training neural networks. Dropout [30] injects noise into feature space by randomly zeroing the activation function to avoid overfitting. Besides, DropConnect [37], Spatial Dropout [33], Droppath [22], DropBlock [11] and Weighted Channel Dropout [16] were also proposed as variants of Dropout. Besides, Batch Normalization [17] improves the gradient propagation through network by normalizing the input for each layer.

Data augmentation: Data augmentation generates virtual training examples in the vicinity of the given training dataset to improve the generalization performance of network. Random cropping and horizontal flipping operatings [20] are the most commonly used data augmentation techniques. By randomly removing contiguous sections of input images, Cutout [6] improves the robustness of network. Random Erasing [41] randomly selects a rectangle region in an image and erases its pixels with random values. Hide-and-Seek [26] and GridMask [4] can be regarded as upgraded versions of Cutout. AutoAugment [5] improves the inception-preprocess using reinforcement learning to search existing policies for the optimal combination.

Mixed Sample Data Augmentation (MSDA): Mixed Sample Data Augmentation has received increasing attention in recent years. Input mixup [40] creates virtual training examples by linearly interpolating two input data and corresponding one-hot labels. Manifold mixup [34] is the variance of mixup, which encourages neural networks to predict less confidently on interpolations of hidden representations. Random image cropping and patching randomly [32] crops four images and patches them to create a new training image. Inspired by Cutout and Mixup, CutMix [38] cut patches and pasted among training images. Based on CutMix, Attentive CutMix [36], FMix [13] and Puzzle Mix [18] aim to capture the most important region(s) of one image and paste it(them) on another one. Cut-Thumbnail can be perfectly integrated with MSDA, because Cut-Thumbnail can introduce most semantics of one image to another image with occupying a small area of it.

6 CONCLUSION

We propose a simple, general and effective data augmentation method named Cut-Thumbnail, which is the first data augmentation method that introduces the idea of thumbnail to data augmentation strategy. We reduce an image to a small size and put it on itself or another image. Different strategies are designed to verify the effectiveness of the thumbnail, and finally Mixed Single Thumbnail works best on different visual tasks. On the ImageNet dataset, Cut-Thumbnail increases the baseline by 2.89%. In fine-grained image classification, Cut-Thumbnail increases the accuracy of ResNet-50 from 85.31% to 87.76% on CUB-200-2011. In the task of Pascal VOC object detection, we improve the baseline by 2.02% on RetinaNet. Extensive experiments have proved that Cut-Thumbnail makes the network better learn shape information and is suitable for different networks, datasets, and tasks. For future work, we plan to find better strategies and hyper-parameters for Cut-Thumbnail using reinforcement learning and apply Cut-Thumbnail to more types of visual tasks.

REFERENCES

[1] Pedro Ballester and Ricardo Araujo. 2016. On the performance of GoogLeNet and AlexNet applied to sketches. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[2] Wieland Brendel and Matthias Bethge. 2019. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760* (2019).

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

pattern analysis and machine intelligence 40, 4 (2017), 834–848.

[4] Pengguang Chen. 2020. GridMask data augmentation. *arXiv preprint arXiv:2001.04086* (2020).

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

[6] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* (2017).

[7] Gil Diesendruck and Paul Bloom. 2003. How specific is the shape bias? *Child development* 74, 1 (2003), 168–178.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 2 (2010), 303–338.

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2017. Texture and art with deep neural networks. *Current opinion in neurobiology* 46 (2017), 178–186.

[10] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* (2018).

[11] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. 2018. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*. 10727–10737.

[12] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[13] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, and Adam Prügel-Bennett Jonathon Hare. 2020. FMix: Enhancing Mixed Sample Data Augmentation. *arXiv preprint arXiv:2002.12047* (2020).

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Saihui Hou and Zilei Wang. 2019. Weighted channel dropout for regularization of deep convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8425–8432.

[17] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[18] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. 2020. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. *arXiv preprint arXiv:2009.06962* (2020).

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

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[21] Barbara Landau, Linda B Smith, and Susan S Jones. 1988. The importance of shape in early lexical learning. *Cognitive development* 3, 3 (1988), 299–321.

[22] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648* (2016).

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[26] Niels Provos and Peter Honeyman. 2003. Hide and seek: An introduction to steganography. *IEEE security & privacy* 1, 3 (2003), 32–44.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.

[29] Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Yadong Mu, and Jingdong Wang. 2020. Informative dropout for robust representation learning: A shape-bias perspective. *arXiv preprint arXiv:2008.04254* (2020).

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[32] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. 2018. Ricap: Random image cropping and patching data augmentation for deep cnns. In *Asian Conference on Machine Learning*. 786–798.

[33] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 648–656.

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

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

[36] Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. 2020. Attentive Cutmix: An Enhanced Data Augmentation Approach for Deep Learning Based Image Classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3642–3646.

[37] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. Regularization of neural networks using dropconnect. In *International conference on machine learning*. 1058–1066.

[38] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*. 6023–6032.

[39] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

[40] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[41] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random Erasing Data Augmentation.. In *AAAI*. 13001–13008.

[42] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.