学习汇报

汇 报 人：高金彤

2022.12.8

The Majority Can Help the Minority: Context-rich Minority Oversampling for Long-tailed Classification

Seulki Park[1*] Youngkyu Hong[2] Byeongho Heo[2] Sangdoo Yun[2] Jin Young Choi[1]

[1]ASRI, ECE., Seoul National University [2]NAVER AI Lab

seulki.park@snu.ac.kr, yk.hong@kaist.ac.kr, {bh.heo, sangdoo.yun}@navercorp.com, jychoi@snu.ac.kr

Introduction

长尾分类的根本问题：类数据不平衡数据集由于缺乏少数类的数据，分类器的泛化性能变差

本篇文章的问题：少样本背景单一

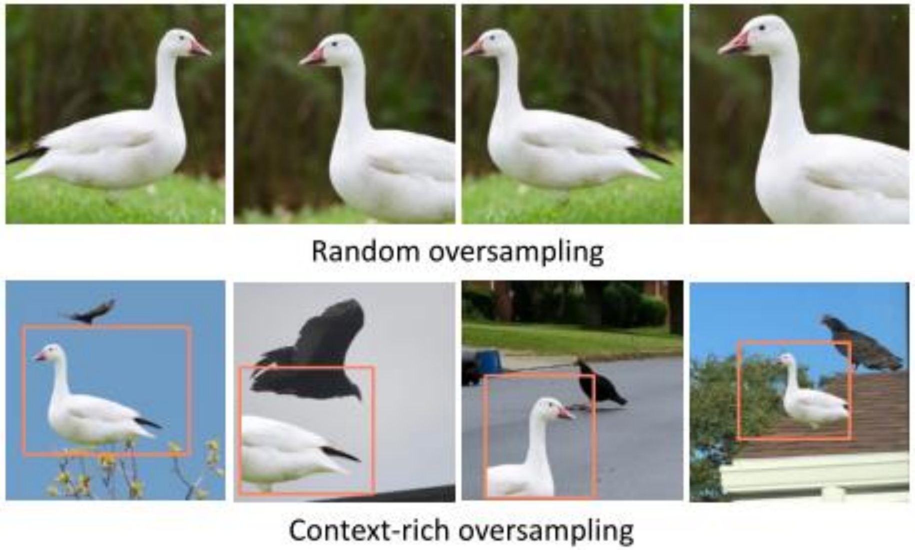
关键思路：少样本多样化（多数类帮助少数类——提供背景实现多样性）

The Majority Can Help the Minority

(少数群体过采样方法，利用多样本的丰富图像作为背景来增强少样本的多样化)

提出：利用Cutmix混合手段生成新的以少样本为中心的图像，具有多样本的背景。

主要创新点：少样本加权分布Q、权重值、即插即用、计算成本低



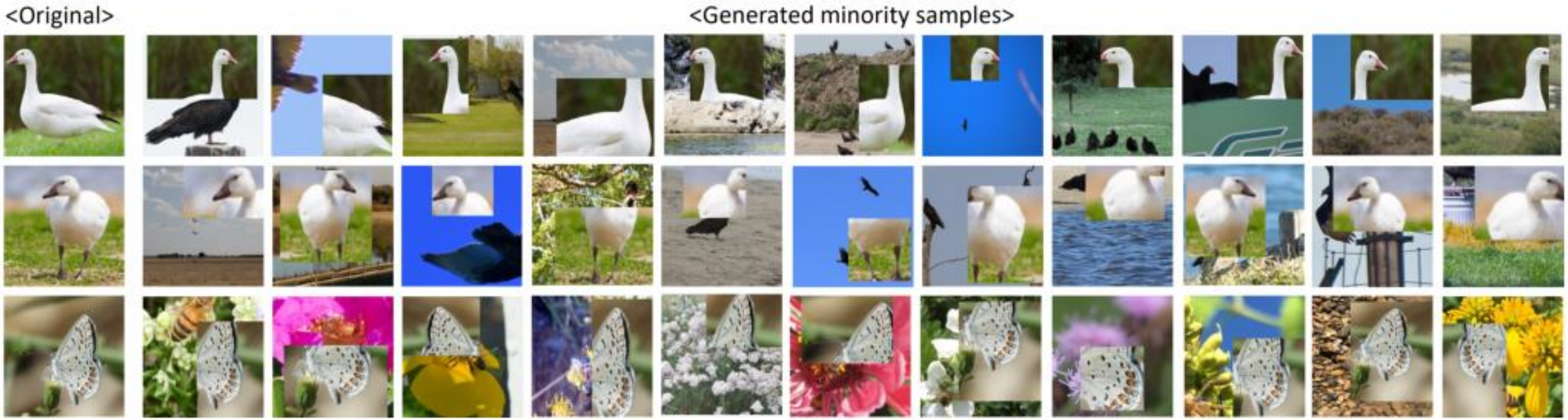


Figure 2. **A display of the minority images generated by CMO** (minority classes: the snow goose and the Acmon blue (butterfly)). We randomly choose generated images for each original image. Our method is able to generate context-rich minority samples that have diverse contexts. For example, while the original 'snow goose' class contains only images of a 'snow goose' on grass, the generated images have various contexts such as the sky, the sea, the sand, and a flock of crows. These generated images enable the model to learn a robust representation of minority classes.

给定一个来自少数类的原始图像，该对象被裁剪成各种大小，并粘贴到来自多数类的各种图像上。然后，我们可以创建具有更多样化背景的图像(例如，带有天空、道路、屋顶、乌鸦等的“雪鹅”图像)。由于这是多数类和少数类样本的插值，因此在决策边界周围生成多样化的数据，从而提高了少数类的泛化性能。

Related Work—Long-tailed Recognition

Re-weighting methods：为训练样本分配不同的权重（概率）

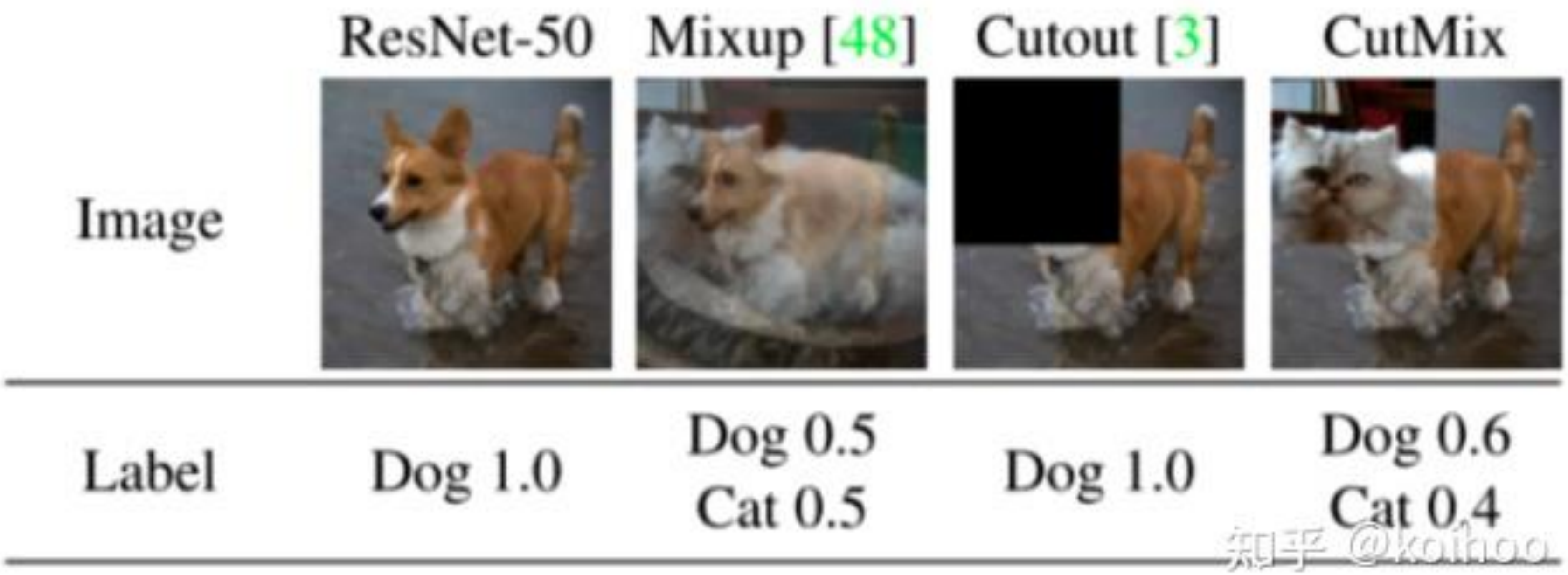
inverse class frequency、CE、LDAM、BS、IB、LADE（loss）

Re-sampling methods：修改训练分布，以降低不平衡水平（图像数量）

1、输入空间：欠采样、过采样（ROS）、合成少样本过采样（SMOTE）、生成对抗少样本过采样（GAMO）
2、特征空间：深度过采样(DOS)、特征空间增强(FSA)和元语义增强(MetaSAug)

Other long-tailed methods：延迟重加权(DRW)，分类器再训练(cRT)，可学习权重缩放(LWS)和Mixup移位标签感知平滑模型

Related Work—Data Augmentation and Mixup Methods



Algorithm

吉林大学

Algorithm 1 Context-rich Minority Oversampling (CMO)

Require: Dataset $\mathcal{D}_{i=1}^{N}$, model parameters θ, P, Q, any loss function $L(\cdot)$.

1: Randomly initialize θ.
2: Sample weighted dataset $\tilde{\mathcal{D}}_{i=1}^{N} \sim Q$.
3: **for** epoch $= 1, \ldots, T$ **do**
4:     **for** batch $i = 1, \ldots, B$ **do**
5:         Draw a mini-batch (x_i^b, y_i^b) from $\mathcal{D}_{i=1}^{N}$
6:         Draw a mini-batch (x_i^f, y_i^f) from $\tilde{\mathcal{D}}_{i=1}^{N}$
7:         $\lambda \sim Beta(\alpha, \alpha)$
8:         $\tilde{x}_i = \mathbf{M} \odot x_i^b + (\mathbf{1} - \mathbf{M}) \odot x_i^f$
9:         $\tilde{y}_i = \lambda y_i^b + (1 - \lambda) y_i^f$
10:         $\theta \leftarrow \theta - \eta \nabla L((\tilde{x}_i, \tilde{y}_i); \theta)$
11:     **end for**
12: **end for**

Algorithm 2 PyTorch-style pseudo-code for CMO

```
# original_loader: data loader from original data distribution
# weighted_loader: data loader from minor-class-weighted distribution
# model: any backbone network such as ResNet or multi-branch networks (RIDE)
# loss: any loss such as CE, LDAM, balanced softmax, RIDE loss

for epoch in Epochs:
    # load a batch for background images from original data dist.
    for x_b, y_b in original_loader:
        # load a batch for foreground from minor-class-weighted dist.
        x_f, y_f = next(weighted_loader)

        # get coordinate for random binary mask
        lambda = np.random.uniform(0,1)
        cx = np.random.randint(W) # W: width of images
        cy = np.random.randint(H) # H: height of images
        bbx1 = np.clip(cx - int(W * np.sqrt(1. - lambda))//2,0,W)
        bbx2 = np.clip(cx + int(W * np.sqrt(1. - lambda))//2,0,W)
        bby1 = np.clip(cy - int(H * np.sqrt(1. - lambda))//2,0,H)
        bby2 = np.clip(cy + int(H * np.sqrt(1. - lambda))//2,0,H)

        # get minor-oversampled images
        x_b[:, :, bbx1:bbx2, bby1:bby2] = x_f[:, :, bbx1:bbx2, bby1:bby2]
        lambda = 1 - ((bbx2 - bbx1) * (bby2 - bby1) / (W * H))# adjust lambda

        # output (x_f is attached to x_b)
        output = model(x_b)

        # loss
        losses = loss(output, y_b) * lambda + loss(output, y_f) * (1. - lambda)

        # optimization step
        losses.backward()
        optimizer.step()
```

Minor-class-weighted Distribution Q

Let n_k be the number of samples in the k-th class, then for the C classes, the total number of samples is $N = \sum_{k=1}^{C} n_k$. Then, the generalized sampling probability for the k-th class can be defined by

$$q(r,k) = \frac{1/n_k^r}{\sum_{k'=1}^{C} 1/n_{k'}^r}, \qquad (2)$$

r = 1 the inverse class frequency

r = 1/2 the smoothed inverse class frequency

```
def get_weighted_sampler(self):
    cls_num_list = self.get_cls_num_list()
    cls_weight = 1.0 / (np.array(cls_num_list) ** self.weighted_alpha)
    cls_weight = cls_weight / np.sum(cls_weight) * len(cls_num_list)
    samples_weight = np.array([cls_weight[t] for t in self.targets])
    samples_weight = torch.from_numpy(samples_weight)
    samples_weight = samples_weight.double()
    print("samples_weight", samples_weight)
    sampler = torch.utils.data.WeightedRandomSampler(samples_weight, len(self.targets), replacement=True)
    return sampler
```

Regularization Effect of CMO

$$\tilde{x} = \mathbf{M} \odot x^b + (\mathbf{1} - \mathbf{M}) \odot x^f$$
$$\tilde{y} = \lambda y^b + (1 - \lambda) y^f, \qquad (1)$$

缓解过度置信度问题：公式1生成的标签会惩罚过度自信的输出，类似于标签平滑正则化

实验设置与结果—datasets

Table 1. **Summary of datasets**. The imbalance ratio ρ is defined by $\rho = \max_k\{n_k\}/\min_k\{n_k\}$, where n_k is the number of samples in the k-th class.

Dataset	# of classes	# of training	Imbalance ratio
CIFAR-100-LT	100	50K	{10, 50, 100}
ImageNet-LT	1,000	115.8K	256
iNaturalist 2018	8,142	437.5K	500

实验设置与结果—datasets

不平衡比率=100（最大类的数量/最小类的数量=500/5=100）：一共有10847张图像
imb_factor=0.01
[500, 477, 455, 434, 415, 396, 378, 361, 344, 328, 314, 299, 286, 273,

不平衡比率=10（最大类的数量/最小类的数量=500/50=10）：一共有19573张图像
imb_factor=0.1
[500, 488, 477, 466, 455, 445, 434, 424, 415, 405, 396, 387, 378, 369,
361, 352, 344, 336, 328, 321, 314, 306, 299, 292, 286, 279, 273, 266,
260, 254, 248, 243, 237, 232, 226, 221, 216, 211, 206, 201, 197, 192,
188, 183, 179, 175, 171, 167, 163, 159, 156, 152, 149, 145, 142, 139,
135, 132, 129, 126, 123, 121, 118, 115, 112, 110, 107, 105, 102, 100, 98,
95, 93, 91, 89, 87, 85, 83, 81, 79, 77, 75, 74, 72, 70, 69, 67, 66, 64,
63, 61, 60, 58, 57, 56, 54, 53, 52, 51, 50]

287, 276, 265, 255, 245, 235, 226, 218, 209, 201, 193, 186, 178, 172,
165, 158, 152, 146, 141, 135, 130, 125, 120, 115, 111, 107, 102, 98, 95,
91, 87, 84, 81, 78, 75, 72, 69, 66, 64, 61, 59, 56, 54, 52, 50, 48, 46,
44, 43, 41, 39, 38, 36, 35, 34, 32, 31, 30, 29, 27, 26, 25, 24, 23, 22,
22, 21, 20, 19, 18, 18, 17, 16, 16, 15, 14, 14, 13, 13, 12, 12, 11, 11,
10, 10, 10]

实验设置与结果—CIFAR-100-LT

设备：	单个NVIDIA GTX 1080Ti GPU
图片大小	32*32
框架与优化器：	Pytorch、SGD(momentum:0.9 weight decay:2*10**-4)
Backbone	Resnet-32
数据增强：	每边填充4个像素，并将水平翻转或随机裁剪应用到32×32大小
学习率：	初始值为0.1，在第160和180个时期*0.01 在前五个epoch使用了学习率的线性预热linear warm-up of the learning rate
Epoch, Batchsize	200, 128
评价指标：	Top1 Accuracy(overall, many, medium, few)
备注：	为了在原始输入空间中微调模型，我们在最后三个阶段关闭了CMO。 我发现在前3个epoch也关闭了CMO

实验设置与结果—ImageNet-LT

设备：	4个NVIDIA GTX 1080Ti GPU
图片大小	224*224
框架与优化器：	Pytorch、SGD(momentum:0.9)
Backbone	Resnet-50
数据增强：	水平翻转、颜色抖动
学习率：	初始值为0.1，在第60和80个时期*0.1
Epoch, Batchsize	100, 256
评价指标：	Top1 Accuracy(overall, many, medium, few)
备注：	为了在原始输入空间中微调模型，我们在最后三个阶段关闭了CMO。 我发现在前3个epoch也关闭了CMO

实验设置与结果—iNaturalist 2018

吉林大学

设备：	8个Tesla V100 GPU
框架与优化器：	Pytorch、SGD(momentum:0.9)
Backbone	Resnet-50、101、152和Wide ResNet-50
数据增强：	水平翻转、颜色抖动
学习率：	初始值为0.1，在第75和160个时期*0.1
Epoch, Batchsize	200, 512
评价指标：	Top1 Accuracy(overall, many, medium, few)
备注：	实验在NAVER智能机器学习（NSML）平台上实施和评估 为了在原始输入空间中微调模型，我们在最后三个阶段关闭了CMO。 我发现在前3个epoch也关闭了CMO

对比实验

Table 2. **State-of-the-art comparison on CIFAR-100-LT dataset.** Classification accuracy (%) for ResNet-32 architecture on CIFAR-100-LT with different imbalance ratios. * and † are from the origi

Table 13. **Comparison against baselines on CIFAR-100-LT** Results with classification accuracy (%) of ResNet-32. The best results are marked in bold.

Imbalance ratio	50				10			
Method	Vanilla	+ROS [47]	+Remix [7]	+CMO	Vanilla	+ROS [47]	+Remix [7]	+CMO
CE	44.0 (+0.0)	39.7 (-4.3)	45.0 (+1.0)	**48.3 (+4.3)**	56.4 (+0.0)	55.6 (-0.8)	58.7 (+2.3)	**59.5 (+3.1)**
CE-DRW [5]	45.6 (+0.0)	41.3 (-4.3)	49.5 (+3.9)	**50.9 (+5.3)**	57.9 (+0.0)	56.4 (-1.5)	59.2 (+1.3)	**61.7 (+3.8)**
LDAM-DRW [5]	47.9 (+0.0)	38.3 (-9.6)	48.8 (+0.9)	**51.7 (+3.8)**	57.3 (+0.0)	53.9 (-3.4)	55.9 (-1.4)	**58.4 (+1.1)**
RIDE [49]	51.4 (+0.0)	31.3 (-20.1)	47.9 (-3.5)	**53.0 (+1.6)**	59.8 (+0.0)	49.4 (-10.4)	59.5 (-0.3)	**60.2 (+0.4)**

LT (Im-
Net-32.

MiSLAS [56]*	47.0	52.3	**63.2**
CE + CMO	43.9	48.3	59.5
CE-DRW + CMO	47.0	50.9	61.7
LDAM-DRW + CMO	47.2	51.7	58.4
BS + CMO	46.6	51.4	62.3
RIDE (3 experts) + CMO	**50.0**	**53.0**	60.2

对比实验

Table 4. **State-of-the-art comparison on ImageNet-LT.** Classification accuracy (%) of ResNet-50 with state-of-the-art methods trained for 90 or 100 epochs. "∗" and "†" denote the results are from the original papers, and [25], respectively. The best results are marked in bold.

	All	Many	Med	Few
Cross Entropy (CE)†	41.6	64.0	33.8	5.8
Decouple-cRT [25]†	47.3	58.8	44.0	26.1
Decouple-LWS [25]†	47.7	57.1	45.2	29.3
Remix [7]	48.6	60.4	46.9	30.7
LDAM-DRW [5]	49.8	60.4	46.9	30.7
CE-DRW	50.1	61.7	47.3	28.8
Balanced Softmax (BS) [40]	51.0	60.9	48.8	32.1
Causal Norm [46]*	51.8	62.7	48.8	31.6
RIDE (3 experts) [49]*	54.9	66.2	51.7	34.9
RIDE (4 experts) [49]*	55.4	66.2	52.3	36.5
CE + CMO	49.1	**67.0**	42.3	20.5
CE-DRW + CMO	51.4	60.8	48.6	35.5
LDAM-DRW + CMO	51.1	62.0	47.4	30.8
BS + CMO	52.3	62.0	49.1	**36.7**
RIDE (3 experts) + CMO	**56.2**	66.4	**53.9**	35.6

Table 5. **Comparison against baselines on ImageNet-LT.** Classification accuracy (%) of ResNet-50.

	Vanilla	+Remix [7]	+CMO
CE	41.6 (+0.0)	41.7 (+0.1)	**49.1 (+7.5)**
CE-DRW [5]	50.1 (+0.0)	48.6 (-1.5)	**51.4 (+1.3)**
Balanced Softmax [40]	51.0 (+0.0)	49.2 (-1.8)	**52.3 (+1.3)**

对比实验

吉林大学

Table 7. **State-of-the-art comparison on iNaturalist2018.** Classification accuracy (%) of ResNet-50 on iNaturalist2018. “$*$” and “†” indicate the results from the original paper and [57], respectively. RIDE [49] was trained for 100 epochs.

	All	Many	Med	Few
Cross Entropy (CE)	61.0	73.9	63.5	55.5
IB Loss [38]*	65.4	-	-	-
FSA [8]*	65.9	-	-	-
LDAM-DRW [5]†	66.1	-	-	-
Decouple-cRT [25]*	68.2	73.2	68.8	66.1
Decouple-LWS [25]*	69.5	71.0	69.8	68.8
BBN [57]*	69.6	-	-	-
Balanced Softmax [40]	70.0	70.0	70.2	69.9
LADE [20]*	70.0	-	-	-
Remix [7]*	70.5	-	-	-
MiSLAS [56]*	71.6	73.2	72.4	70.4
RIDE (3 experts) [49]*	72.2	70.2	72.2	72.7
RIDE (4 experts) [49]*	72.6	70.9	72.4	**73.1**
CE + CMO	68.9	**76.9**	69.3	66.6
CE-DRW + CMO	70.9	68.2	70.2	72.2
LDAM-DRW + CMO	69.1	75.3	69.5	67.3
BS + CMO	70.9	68.8	70.0	72.3
CE-DRW + CMO + LAS [56]	71.8	69.6	72.1	71.9
RIDE (3 experts) + CMO	**72.8**	68.7	**72.6**	**73.1**

Table 8. **Results on large architectures.** Classification accuracy (%) of large backbone networks on iNaturalist 2018. The results are copied from [8].

Method	ResNet-50	Wide ResNet-50	ResNet-101	ResNet-152
CE	61.0	-	65.2	66.2
FSA [8]	65.9	-	68.4	69.1
CMO	**70.9**	**71.9**	**72.4**	**72.6**

对比实验

Imbalance ratio	100	50	10
BS*	50.8	54.2	63.0
PaCo [11]*	**52.0**	56.0	64.2
BS + CMO	51.7	**56.7**	**65.3**

Table 14. **Classification Accuracy on CIFAR-100-LT with different imbalance ratios.** We train ResNet-32 with AutoAugment [9] in 400 epochs. $*$ is from [11] The best results are marked in bold.

Table 6. **Results on longer training epochs with RandAugment [10].** Classification accuracy (%) of ResNet-50 on ImageNet-LT. "$*$" denotes the results from [11].

	All	Many	Med	Few
BS*	55.0	66.7	52.9	33.0
PaCo [11]*	57.0	65.0	**55.7**	38.2
BS + CMO	**58.0**	**67.0**	55.0	**44.2**

	All	Many	Med	Few
BS*	71.8	**72.3**	72.6	71.7
PaCo [11]*	73.2	70.3	73.2	73.6
BS + CMO	**74.0**	71.9	**74.2**	**74.2**

Table 15. **Classification Accuracy on iNaturalist2018.** We train ResNet-50 for 400 epochs with RandAugment [10]. "$*$" indicates the results are from [11]. The best results are marked in bold.

Impact of α

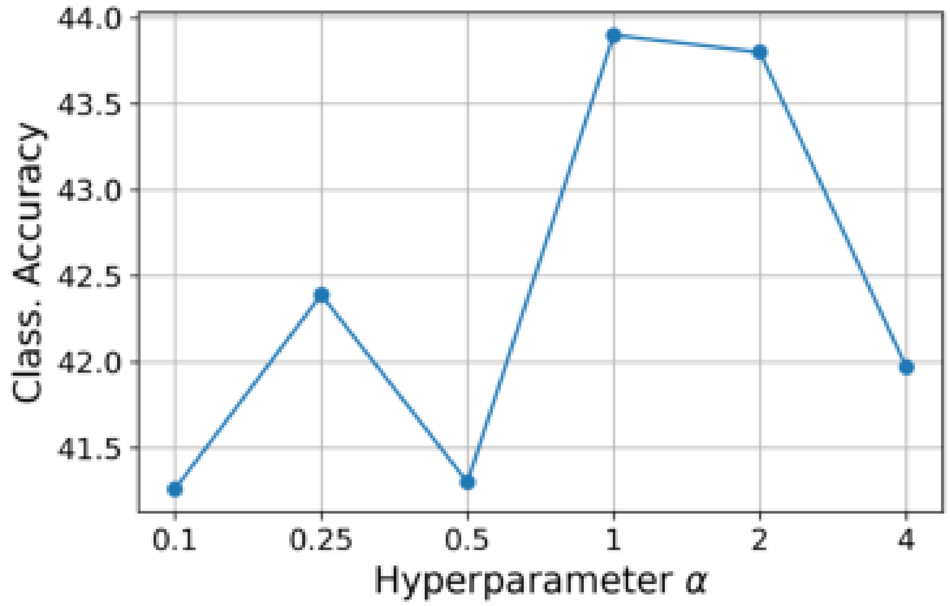


Figure 3. **Impact of** α on CIFAR-100-LT with an imbalance ratio of 100.

Analysis

Is the distribution for augmenting images important?（比Cutmix好的原因）

How to choose the appropriate probability distribution Q? （Q的r选择）

Why should we oversample only for the foreground samples?（只在前景使用的原因）

Comparison with other minority augmentations.（常见数据增强的手段）

Is the distribution for augmenting images important?

吉林大学

Table 9. **Comparison with CutMix** using cross-entropy loss.

	All	Many	Med	Few
CIFAR-100-LT				
CutMix	35.6	71.0	37.9	4.9
CMO	43.9	70.4	42.5	14.4
ImageNet-LT				
CutMix	45.5	68.6	38.1	8.1
CMO	49.1	67.0	42.3	20.5

How to choose the appropriate probability distribution Q?

Table 10. **Impact of different Q sampling distributions.** Results on CIFAR-100-LT (imbalance ratio=100) according to different Q sampling probabilities.

	All	Many	Med	Few
$q(1/2, k)$	42.6	**71.6**	42.1	9.5
$q(1, k)$	**43.9**	70.4	**42.5**	**14.4**
$q(2, k)$	40.1	67.2	36.7	12.3
$E(k)$ [12]	39.5	70.4	38.0	4.7

Why should we oversample only for the foreground samples?

吉林大学

Table 11. **Ablation study.** Results from variants of CMO with ResNet-32 on imbalanced CIFAR-100; imbalance ratio of 100.

	All	Many	Med	Few
Cross Entropy (CE)	38.6	65.3	37.6	8.7
CE + CMO $_{minor}$	37.9	58.3	40.4	11.2
CE + CMO $_{back}$	40.1	64.7	40.2	11.3
CE + CMO	**43.9**	**70.4**	**42.5**	**14.4**
LDAM [5]	41.7	61.4	42.2	18.0
LDAM + CMO $_{minor}$	31.7	50.2	33.2	8.4
LDAM + CMO $_{back}$	44.2	59.2	46.6	24.0
LDAM + CMO	**47.2**	**61.5**	**48.6**	**28.8**

这是因为，使用CutMix方法，前景图像中的对象很有可能与背景图像重叠。因此，我们可以预期背景图像中关于少样本的信息会丢失，从而导致有限的性能提升。

Comparison with other minority augmentations?

吉林大学

Table 12. **Data augmentation methods.** Comparisons between augmentation methods for generating new minority samples on CIFAR-100-LT with an imbalance ratio of 100.

	All	Many	Med	Few
CMO w/ Gaussian Blur	31.1	54.7	28.8	6.2
CMO w/ Color Jitter	34.7	58.9	34.4	6.8
CMO w/ Mixup	38.0	54.8	40.2	**15.9**
CMO w/ CutMix	**43.9**	**70.4**	**42.5**	14.4

产生的想法和问题

吉林大学

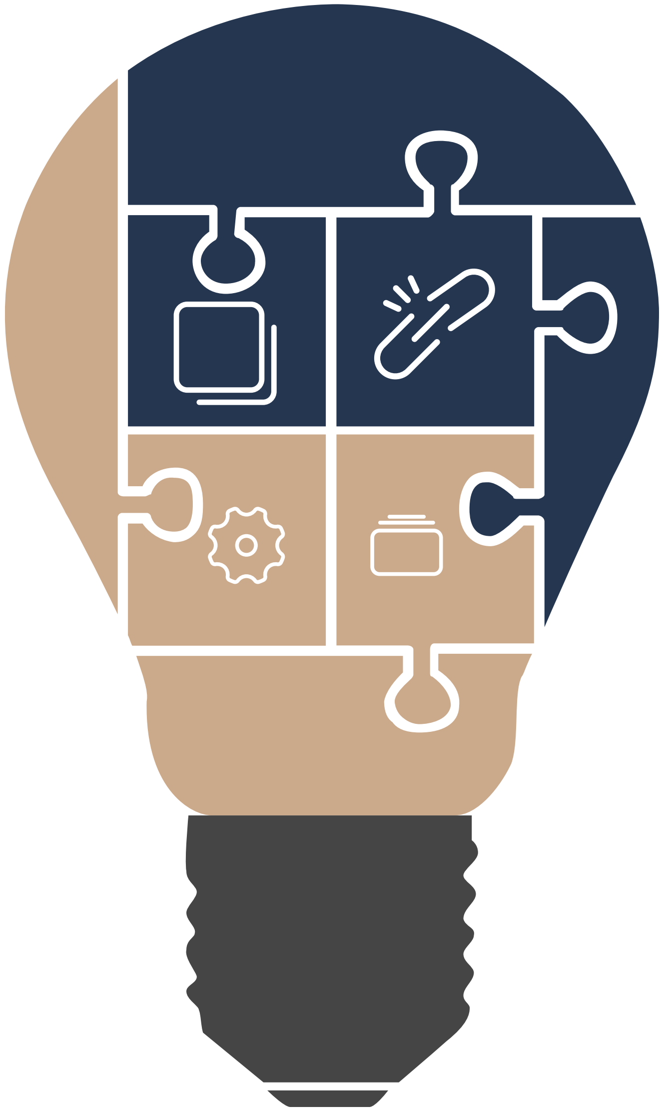
Q1 少样本和多样本的效果难以实现同时上升。

Q2 硬件问题

I1 分组：500抽5张+少5张->一组

500中初概率相同，被选中该图像的概率下降，未选中概率上升

I2 通过损失函数（多损失、中损失和少损失）+GAN



THANKS

恳请老师批评指正

汇 报 人： 高金彤

2022.12.8