CutMix:
Regularization Strategy to Train Strong Classifiers with Localizable Features

Sangdoo Yun, Dongyoon Han, Seongjoon Oh,
Sanghyuk Chun, Junsuk Choe, Youngjoon Yoo
- Clova AI Reseach

박태우

Introduction

- Input 이미지의 작은 부분에 CNN이 너무 과한 가중치를 두는 것을 막기 위해 다양한 방법들이 사용되어져 왔음. (activations 또는 image의 일부분을 랜덤하게 dropout 함.) ex) Mixup, Cutout

- 본 논문은 image A에 다른 image B를 얹어 training data로 사용하는 CutMix를 제안함.

Introduction

- CutMix 전에 사용되던 Mixup, Cutout의 경우 classification의 성능은 증가했지만

localization 또는 object dectection 성능이 낮아지는 결과를 가져왔음.

- CutMix는 이러한 문제점들을 보완하고자 함.

	ResNet-50	Mixup [48]	Cutout [3]
Image			
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)

CutMix Algorithm

- Image A, B가 있을 때 B의 일부분을 A에 붙여 새로운 training data를 만듬.
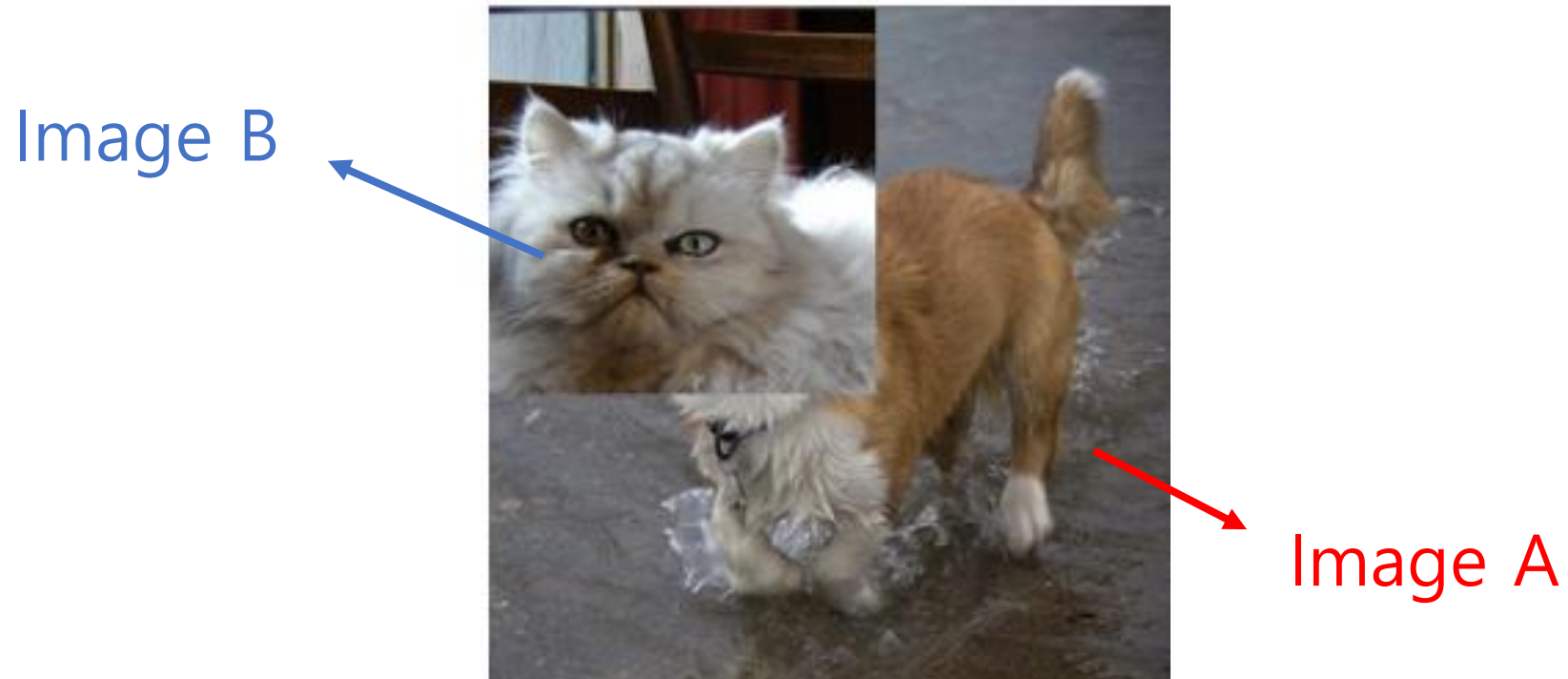
Image B

Image A

- 새로운 데이터는 image $\tilde{x}$ 와 label $\tilde{y}$ 을 다음과 같이 표현함.

$$\tilde{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B$$

$$\tilde{y} = \lambda y_A + (1 - \lambda) y_B,$$

- M은 binary mask로써 image A의 영역을 1로 표현하고 이외에는 0으로 표현한다.

- λ 는 전체 이미지 크기에서 A가 차지하는 비율을 나타냄.

$$\lambda = \frac{\square}{\square}$$

$$= 12 \,/\, 16$$

$$= 0.75$$

CutMix Algorithm

Model training code using CutMix

```
# generate mixed sample
lam = np.random.beta(args.beta, args.beta)
rand_index = torch.randperm(input.size()[0]).cuda()

target_a = target
target_b = target[rand_index]

bbx1, bby1, bbx2, bby2 = rand_bbox(input.size(), lam)
input[:, :, bbx1:bbx2, bby1:bby2] = input[rand_index, :, bbx1:bbx2, bby1:bby2]
# adjust lambda to exactly match pixel ratio
lam = 1 - ((bbx2 - bbx1) * (bby2 - bby1) / (input.size()[-1] * input.size()[-2]))
# compute output
output = model(input)
loss = criterion(output, target_a) * lam + criterion(output, target_b) * (1. - lam)
```

input : batch 내 모든 image
target : batch 내 모든 label (class마다 값 존재)

랜덤으로 B image 선택

모든 image에
B의 일부분 붙여넣기

A가 차지하는 비중만큼
lamba 계산

모든 image의 softmax 값 계산

합성된 image의 softmax < - > A label, B label 의 cross entropy 값 각각 계산.
그리고 각각 전체 이미지에서 해당되는 비율만큼 곱함. (λ , 1 – λ)

**즉, 합성 image에 새로운 label 값 할당이 아니라,
image 생성과 동시에 training model의 loss를 계산.**

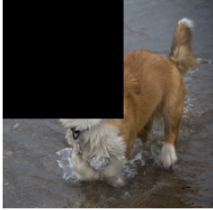| Image	ResNet-50	Mixup [48]	Cutout [3]	CutMix
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)	**78.6** (+2.3)
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)	**47.3** (+1.0)
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)	**76.7** (+1.1)

Table 1: Overview of the results of Mixup, Cutout, and our CutMix on ImageNet classification, ImageNet localization, and Pascal VOC 07 detection (transfer learning with SSD [24] finetuning) tasks. Note that CutMix significantly improves the performance on various tasks.
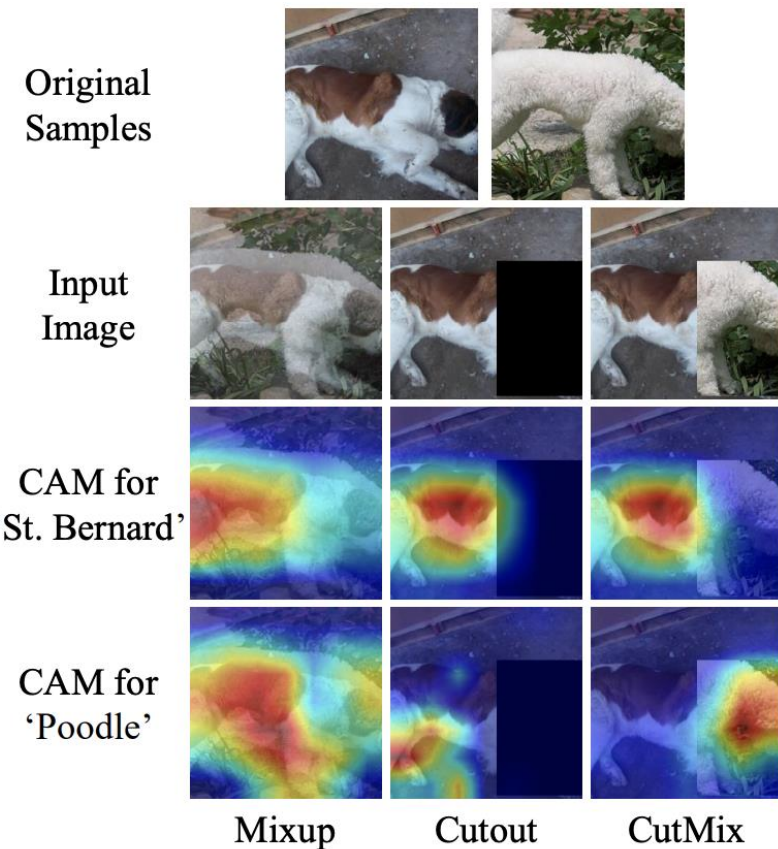

Figure 1: Class activation mapping (CAM) [52] visualizations on 'Saint Bernard' and 'Miniature Poodle' samples using various augmentation techniques. From top to bottom rows, we show the original images, input augmented image, CAM for class 'Saint Bernard', and CAM for class 'Miniature Poodle', respectively. Note that CutMix can take advantage of the mixed region on image, but Cutout cannot.
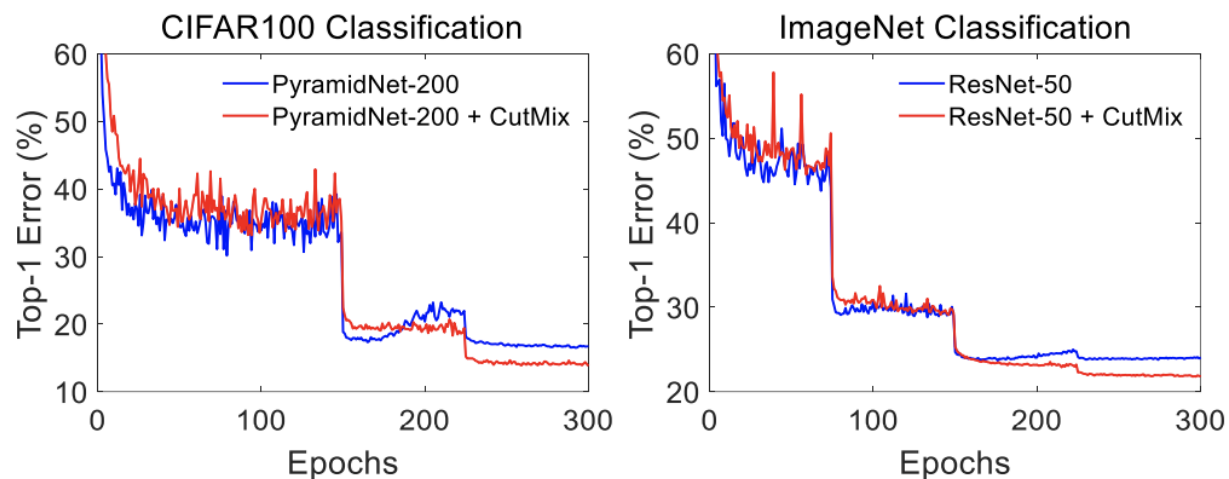
Figure 2: Top-1 test error plot for CIFAR100 (left) and Im-ageNet (right) classification. Cutmix achieves lower test er-rors than the baseline at the end of training.

Model	# Params	Top-1 Err (%)	Top-5 Err (%)
ResNet-101 (Baseline) [12]	44.6 M	21.87	6.29
ResNet-101 + Cutout [3]	44.6 M	20.72	5.51
ResNet-101 + Mixup [48]	44.6 M	20.52	5.28
ResNet-101 + CutMix	44.6 M	**20.17**	**5.24**
ResNeXt-101 (Baseline) [45]	44.1 M	21.18	5.57
ResNeXt-101 + CutMix	44.1 M	**19.47**	**5.03**

Table 4: Impact of CutMix on ImageNet classification for ResNet-101 and ResNext-101.

Backbone Network	ImageNet Cls Top-1 Error (%)	Detection		Image Captioning	
		SSD [24] (mAP)	Faster-RCNN [30] (mAP)	NIC [43] (BLEU-1)	NIC [43] (BLEU-4)
ResNet-50 (Baseline)	23.68	76.7 (+0.0)	75.6 (+0.0)	61.4 (+0.0)	22.9 (+0.0)
Mixup-trained	22.58	76.6 (-0.1)	73.9 (-1.7)	61.6 (+0.2)	23.2 (+0.3)
Cutout-trained	22.93	76.8 (+0.1)	75.0 (-0.6)	63.0 (+1.6)	24.0 (+1.1)
CutMix-trained	21.40	**77.6 (+0.9)**	**76.7 (+1.1)**	**64.2 (+2.8)**	**24.9 (+2.0)**

Table 10: Impact of CutMix on transfer learning of pretrained model to other tasks, object detection and image captioning.

References

https://github.com/clovaai/CutMix-PyTorch