CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features – Supplementary Material –

Sangdoo Yun[1] Dongyoon Han[1] Seong Joon Oh[2] Sanghyuk Chun[1]
Junsuk Choe[1,3] Youngjoon Yoo[1]

[1]Clova AI Research, NAVER Corp.
[2]Clova AI Research, LINE Plus Corp.
[3]Yonsei University

A. CutMix Algorithm

We present the code-level description of CutMix algorithm in Algorithm A1. N, C, and K denote the size of minibatch, channel size of input image, and the number of classes. First, CutMix shuffles the order of the minibatch input and target along the first axis of the tensors. And the lambda and the cropping region (x1,x2,y1,y2) are sampled. Then, we mix the input and input_s by replacing the cropping region of input to the region of input_s. The target label is also mixed by interpolating method.

Note that CutMix is easy to implement with few lines (from line 4 to line 15), so is very practical algorithm giving significant impact on a wide range of tasks.

B. Weakly-supervised Object Localization

We describe the training and evaluation procedure of weakly-supervised object localization in detail.

Network modification: Basically weakly-supervised object localization (WSOL) has the same training strategy as image classification does. Training WSOL is starting from ImageNet-pretrained model. From the base network structures, VGG-16 and ResNet-50 [3], WSOL takes larger spatial size of feature map 14×14 whereas the original models has 7×7. For VGG network, we utilize VGG-GAP, which is a modified VGG-16 introduced in [16]. For ResNet-50, we modified the final residual block (`layer4`) to have no stride ($= 1$), which originally has stride 2.

Since the network is modified and the target dataset could be different from ImageNet [10], the last fully-connected layer is randomly initialized with the final output dimension of 200 and 1000 for CUB200-2011 [13] and ImageNet, respectively.

Input image transformation: For fair comparison, we used the same data augmentation strategy except Mixup, Cutout, and CutMix as the state-of-the-art WSOL methods do [11, 15]. In training, the input image is resized to 256×256 size and randomly cropped 224×224 size images are used to train network. In testing, the input image is resized to 256×256, cropped at center with 224×224 size and used to validate the network, which called single crop strategy.

Estimating bounding box: We utilize class activation mapping (CAM) [16] to estimate the bounding box of an object. First we compute CAM of an image, and next, we decide the foreground region of the image by binarizing the CAM with a specific threshold. The region with intensity over the threshold is set to 1, otherwise to 0. We use the threshold as a specific rate σ of the maximum intensity of the CAM. We set σ to 0.15 for all our experiments. From the binarized foreground map, the tightest box which can cover the largest connected region in the foreground map is selected to the bounding box for WSOL.

Evaluation metric: To measure the localization accuracy of models, we report top-1 localization accuracy (Loc), which is used for ImageNet localization challenge [10]. For top-1 localization accuracy, intersection-over-union (IoU) between the estimated bounding box and ground truth position is larger than 0.5, and, at the same time, the estimated class label should be correct. Otherwise, top-1 localization accuracy treats the estimation was wrong.

B.1. CUB200-2011

CUB-200-2011 dataset [13] contains over 11 K images with 200 categories of birds. We set the number of training epochs to 600. For ResNet-50, the learning rate for the last fully-connected layer and the other were set to 0.01 and 0.001, respectively. For VGG network, the learning rate for the last fully-connected layer and the other were set to 0.001 and 0.0001, respectively. The learning rate is decaying by

Algorithm A1 Pseudo-code of CutMix

```
for each iteration do
    input, target = get_minibatch(dataset)          ▷ input is N×C×W×H size tensor, target is N×K size tensor.
    if mode == training then
        input_s, target_s = shuffle_minibatch(input, target)          ▷ CutMix starts here.
        lambda = Unif(0,1)
        r_x = Unif(0,W)
        r_y = Unif(0,H)
        r_w = Sqrt(1 - lambda)
        r_h = Sqrt(1 - lambda)
        x1 = Round(Clip(r_x - r_w / 2, min=0))
        x2 = Round(Clip(r_x + r_w / 2, max=W))
        y1 = Round(Clip(r_y - r_h / 2, min=0))
        y2 = Round(Clip(r_y + r_h / 2, min=H))
        input[:, :, x1:x2, y1:y2] = input_s[:, :, x1:x2, y1:y2]
        lambda = 1 - (x2-x1)*(y2-y1)/(W*H)          ▷ Adjust lambda to the exact area ratio.
        target = lambda * target + (1 - lambda) * target_s          ▷ CutMix ends.
    end if
    output = model_forward(input)
    loss = compute_loss(output, target)
    model_update()
end for
```

the factor of 0.1 at every 150 epochs. We used SGD optimizer, and the minibatch size, momentum, weight decay were set to 32, 0.9, and 0.0001.

B.2. ImageNet dataset

ImageNet-1K [10] is a large-scale dataset for general objects consisting of 13 M training samples and 50 K validation samples. We set the number of training epochs to 20. The learning rate for the last fully-connected layer and the other were set to 0.1 and 0.01, respectively. The learning rate is decaying by the factor of 0.1 at every 6 epochs. We used SGD optimizer, and the minibatch size, momentum, weight decay were set to 256, 0.9, and 0.0001.

C. Transfer Learning to Object Detection

We evaluate the models on the Pascal VOC 2007 detection benchmark [1] with 5 K `test` images over 20 object categories. For training, we use both VOC2007 and VOC2012 `trainval` (VOC07+12).

Finetuning on SSD[1] **[8]:** The input image is resized to 300×300 (SSD300) and we used the basic training strategy of the original paper such as data augmentation, prior boxes, and extra layers. Since the backbone network is changed from VGG16 to ResNet-50, the pooling location `conv4_3` of VGG16 is modified to the output of `layer2` of ResNet-50. For training, we set the batch size, learning rate, and training iterations to 32, 0.001, and 120 K, respectively. The learning rate is decayed by the factor of 0.1 at 80 K and 100 K iterations.

Finetuning on Faster-RCNN[2] **[9]:** Faster-RCNN takes fully-convolutional structure, so we only modify the backbone from VGG16 to ResNet-50. The batch size, learning rate, training iterations are set to 8, 0.01, and 120 K. The learning rate is decayed by the factor of 0.1 at 100 K iterations.

D. Transfer Learning to Image Captioning

MS-COCO dataset [7] contains 120 K `trainval` images and 40 K `test` images. From the base model NIC[3] [12], the backbone model is changed from GoogLeNet to ResNet-50. For training, we set batch size, learning rate, and training epochs to 20, 0.001, and 100, respectively. For evaluation, the beam size is set to 20 for all the experiments. Image captioning results with various metrics are shown in Table A1.

E. Robustness and Uncertainty

In this section, we describe the details of the experimental setting and evaluation methods.

E.1. Robustness

We evaluate the model robustness to adversarial perturbations, occlusion and in-between samples using Ima-

[1] https://github.com/amdegroot/ssd.pytorch

[2] https://github.com/jwyang/faster-rcnn.pytorch

[3] https://github.com/stevehuanghe/image_captioning

	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE	CIDER
ResNet-50 (Baseline)	61.4	43.8	31.4	22.9	22.8	44.7	71.2
ResNet-50 + Mixup	61.6	44.1	31.6	23.2	22.9	47.9	72.2
ResNet-50 + Cutout	63.0	45.3	32.6	24.0	22.6	48.2	74.1
ResNet-50 + CutMix	**64.2**	**46.3**	**33.6**	**24.9**	**23.1**	**49.0**	**77.6**

Table A1: Image captioning results on MS-COCO dataset.

geNet trained models. For the base models, we use ResNet-50 structure and follow the settings in Section 4.1.1. For comparison, we use ResNet-50 trained without any additional regularization or augmentation techniques, ResNet-50 trained by Mixup strategy, ResNet-50 trained by Cutout strategy and ResNet-50 trained by our proposed CutMix strategy.

Fast Gradient Sign Method (FGSM): We employ Fast Gradient Sign Method (FGSM) [2] to generate adversarial samples. For the given image x, the ground truth label y and the noise size ϵ, FGSM generates an adversarial sample as the following

$$\hat{x} = x + \epsilon \, \mathtt{sign} \left(\nabla_x L(\theta, x, y) \right), \tag{1}$$

where $L(\theta, x, y)$ denotes a loss function, for example, cross entropy function. In our experiments, we set the noise scale $\epsilon = 8/255$.

Occlusion: For the given hole size s, we make a hole with width and height equals to s in the center of the image. For center occluded samples, we zeroed-out inside of the hole and for boundary occluded samples, we zeroed-out outside of the hole. In our experiments, we test the top-1 ImageNet validation accuracy of the models with varying hole size from 0 to 224.

In-between class samples: To generate in-between class samples, we first sample $50,000$ pairs of images from the ImageNet validation set. For generating Mixup samples, we generate a sample x from the selected pair x_A and x_B by $x = \lambda x_A + (1 - \lambda) x_B$. We report the top-1 accuracy on the Mixup samples by varying λ from 0 to 1. To generate CutMix in-between samples, we employ the center mask instead of the random mask. We follow the hole generation process used in the occlusion experiments. We evaluate the top-1 accuracy on the CutMix samples by varing hole size s from 0 to 224.

E.2. Uncertainty

Deep neural networks are often overconfident in their predictions. For example, deep neural networks produce high confidence number even for random noise [4]. One standard benchmark to evaluate the overconfidence of the network is Out-of-distribution (OOD) detection proposed by [4]. The authors proposed a threshold-baed detector which solves the binary classification task by classifying in-distribution and out-of-distribution using the prediction of the given network. Recently, a number of reserchs are proposed to enhance the performance of the baseline detector [6, 5] but in this paper, we follow only the baseline detector algorithm without any input enhancement and temperature scaling [6].

Setup: We compare the OOD detector performance using CIFAR-100 trained models described in Section 4.1.2. For comparison, we use PyramidNet-200 model without any regularization method, PyramidNet-200 model with Mixup, PyramidNet-200 model with Cutout and PyramidNet-200 model with our proposed CutMix.

Evaluation Metrics and Out-of-distributions: In this work, we follow the experimental setting used in [4, 6]. To measure the performance of the OOD detector, we report the true negative rate (TNR) at 95% true positive rate (TPR), the area under the receiver operating characteristic curve (AUROC) and detection accuracy of each OOD detector. We use seven datasets for out-of-distribution: TinyImageNet (crop), TinyImageNet (resize), LSUN [14] (crop), LSUN (resize), iSUN, Uniform noise and Gaussian noise.

Results: We report OOD detector performance to seven OODs in Table A2. Overall, CutMix outperforms baseline, Mixup and Cutout. Moreover, we find that even though Mixup and Cutout outperform the classification performance, Mixup and Cutout largely degenerate the baseline detector performance. Especially, for Uniform noise and Gaussian noise, Mixup and Cutout seriously impair the baseline performance while CutMix dramatically improves the performance. From the experiments, we observe that our proposed CutMix enhances the OOD detector performance while Mixup and Cutout produce more overconfident predictions to OOD samples than the baseline.

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Method	TNR at TPR 95%	AUROC	Detection Acc.	TNR at TPR 95%	AUROC	Detection Acc.
	TinyImageNet			TinyImageNet (resize)		
Baseline	43.0 (0.0)	88.9 (0.0)	81.3 (0.0)	29.8 (0.0)	84.2 (0.0)	77.0 (0.0)
Mixup	22.6 (-20.4)	71.6 (-17.3)	69.8 (-11.5)	12.3 (-17.5)	56.8 (-27.4)	61.0 (-16.0)
Cutout	30.5 (-12.5)	85.6 (-3.3)	79.0 (-2.3)	22.0 (-7.8)	82.8 (-1.4)	77.1 (+0.1)
CutMix	**57.1** (+14.1)	**92.4** (+3.5)	**85.0** (+3.7)	**55.4** (+25.6)	**91.9** (+7.7)	**84.5** (+7.5)
	LSUN (crop)			LSUN (resize)		
Baseline	34.6 (0.0)	86.5 (0.0)	79.5 (0.0)	34.3 (0.0)	86.4 (0.0)	79.0 (0.0)
Mixup	22.9 (-11.7)	76.3 (-10.2)	72.3 (-7.2)	13.0 (-21.3)	59.0 (-27.4)	61.8 (-17.2)
Cutout	33.2 (-1.4)	85.7 (-0.8)	78.5 (-1.0)	23.7 (-10.6)	84.0 (-2.4)	78.4 (-0.6)
CutMix	**47.6** (+13.0)	**90.3** (+3.8)	**82.8** (+3.3)	**62.8** (+28.5)	**93.7** (+7.3)	**86.7** (+7.7)
		iSUN				
Baseline		32.0 (0.0)	85.1 (0.0	77.8 (0.0)		
Mixup		11.8 (-20.2)	57.0 (-28.1)	61.0 (-16.8)		
Cutout		22.2 (-9.8)	82.8 (-2.3)	76.8 (-1.0)		
CutMix		**60.1** (+28.1)	**93.0** (+7.9)	**85.7** (+7.9)		
	Uniform			Gaussian		
Baseline	0.0 (0.0)	89.2 (0.0)	89.2 (0.0)	10.4 (0.0)	90.7 (0.0)	89.9 (0.0)
Mixup	0.0 (0.0)	0.8 (-88.4)	50.0 (-39.2)	0.0 (-10.4)	23.4 (-67.3)	50.5 (-39.4)
Cutout	0.0 (0.0)	35.6 (-53.6)	59.1 (-30.1)	0.0 (-10.4)	24.3 (-66.4)	50.0 (-39.9)
CutMix	**100.0** (+100.0)	**99.8** (+10.6)	**99.7** (+10.5)	**100.0** (+89.6)	**99.7** (+9.0)	**99.0** (+9.1)

Table A2: Out-of-distribution (OOD) detection results on TinyImageNet, LSUN, iSUN, Gaussian noise and Uniform noise using CIFAR-100 trained models. All numbers are in percents; higher is better.

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

[4] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[5] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[6] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

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

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[11] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553. IEEE, 2017.

[12] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[13] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[14] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[15] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for

weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, 2018.

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