REGION MIXUP

Saptarshi Saha & Utpal Garain
Indian Statistical Institute Kolkata
{s.saha_r,utpal}@isical.ac.in

ABSTRACT

This paper introduces a simple extension of mixup (Zhang et al., 2018) data augmentation to enhance generalization in visual recognition tasks. Unlike the vanilla mixup method, which blends entire images, our approach focuses on combining regions from multiple images.

1 INTRODUCTION

Mixup (Zhang et al., 2018) is a data augmentation method that trains models on weighted averages of randomly paired training points. The averaging weights are typically sampled from a beta distribution with parameter α, where α ensures that the generated training set remains close to the original dataset. Mixup-generated perturbations may adhere only to the direction towards any arbitrary data point, potentially resulting in suboptimal regularization (Guo et al., 2019). To this end, we propose Region Mixup, an approach emphasizing the integration of regions from multiple images. While various mixup variants (Verma et al., 2018; Kim et al., 2021; Liu et al., 2021) have been proposed to address suboptimal regularization, including those considering convex combinations of more than two points, yet none explicitly strive to interpolate at the level of regions. Closest to our work is CutMix (Yun et al., 2019). However, CutMix does not interpolate regions; instead, it cuts and pastes patches between training images.

2 REGION MIXUP

Let $x \in \mathbb{R}^{W \times H \times C}$ and y represent a training image and its corresponding label, respectively. The objective of region mixup is to create a new training sample $(\tilde{x}, \tilde{y})$ by combining regions from multiple training samples $(x_A, y_A), (x_{B_1}, y_{B_1}), (x_{B_2}, y_{B_2}), \dots, (x_{B_{k^2}}, y_{B_{k^2}})$. The combining operation is defined as follows:

$$\tilde{x} = \sum_{j=1}^{k^2} \lambda_j M_j \odot x_A + (1 - \lambda_j) M_j \odot x_{B_j}, \quad \text{and} \quad \tilde{y} = \frac{1}{k^2} \sum_{j=1}^{k^2} \lambda_j y_A + (1 - \lambda_j) y_{B_j}, \quad (1)$$

where $M_j \in \{0, 1\}^{W \times H}$ denotes a binary mask representing the region to be mixed up from two images x_A and x_{B_j}, and $\sum_{j=1}^{k^2} M_j = \mathbf{1}$. The operation $\odot$ denotes element-wise multiplication. If $k = 1$, we recover standard mixup regularization.

Algorithm 1 Region Mixup at t-th training iteration

Input: Mini-batch $(\mathbf{x}, \mathbf{y})$, classifier f with parameters θ_{t-1}, and model optimizer SGD
1: Sample mixup parameters $\lambda_1, \lambda_2, ..., \lambda_{k^2} \sim Beta(\alpha, \alpha)$
2: $(\mathbf{x}_A, \mathbf{y}_A) = (\mathbf{x}, \mathbf{y})$; $(\mathbf{x}_{B_j}, \mathbf{y}_{B_j}) = $ **RandomPermute**$(\mathbf{x}, \mathbf{y})$ for $j = 1$ to k^2
3: Compute $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ using equation 1
4:
$$\mathcal{L} = CE\Big(f(\mathbf{x}_A), \mathbf{y}_A\Big) + CE\Big(f(\tilde{\mathbf{x}}), \tilde{\mathbf{y}}\Big)$$

$\triangleright$ *CE* is cross-entorpy loss.

5: $\theta_t = \text{SGD}\big(\theta_{t-1}, \frac{\partial \mathcal{L}}{\partial \theta_{t-1}}\big)$
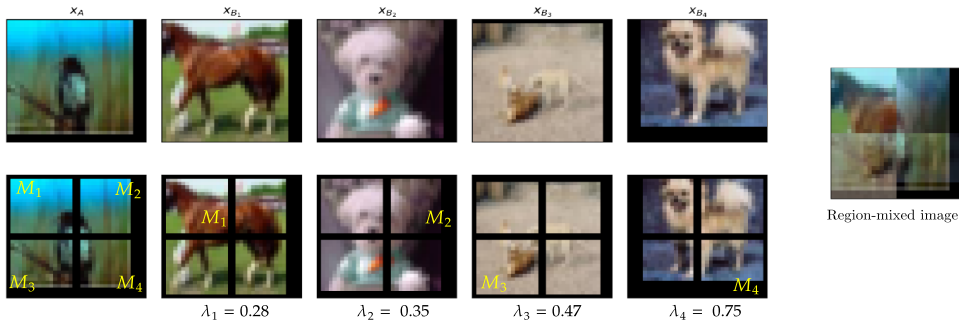**Output:** Updated parameters θ_t

---

Figure 1: Understanding region mixup

Although introducing stochasticity into the region selection process is an intriguing avenue for future exploration, we opted for a straightforward approach in this work. We divide every image into non-overlapping tiles of equal size, forming regions in a grid pattern with dimensions $k \times k$ (see figure 1). In conjunction with the mixup loss, we incorporate the standard cross-entropy loss (highlighted in magenta in the algorithm 1) for classification. Experimentally, we find this combined loss performs better (see Table 2 in Appendix).

3 EXPERIMENTS

We perform image classification experiments on the CIFAR-10, CIFAR-100, and Tiny ImageNet datasets to assess the generalization capabilities of region mixup. In particular, we evaluate Mixup (Zhang et al., 2018), CutMix (Yun et al., 2019) , and Region mixup for the PreAct ResNet-18 (He et al., 2016). All models undergo training on a single Nvidia RTX A5000 using PyTorch Lightning (Falcon & The PyTorch Lightning team, 2019) for 400 epochs on the training set, employing 128 examples per minibatch. Evaluation is carried out on the test set. The training utilizes SGD with momentum, a weight decay of 0.0005, and a step-wise learning rate decay. The learning rates commence at 0.1 and undergo division by 10 after 100 and 150 epochs during the training process. We do not use dropout in these experiments. For all three dataset, each image is zero-padded with two pixels on each side. Subsequently, for CIFAR-10 and CIFAR-100, the resulting image is cropped randomly to generate a new 32×32 image. For Tiny ImageNet, the random cropping process generates a new 64×64 image. Next, we flip the image horizontally with a probability of 50%. We summarize our results in Table 1 and Table 3 (in Appendix). The results are averaged over 3 runs.

Table 1: Test accuracy for the CIFAR and Tiny ImageNet experiments.

Dataset	Model	CutMix	Vanilla Mixup $k = 1$	Region Mixup $k = 2$
CIFAR-10		95.82±0.19	95.89±0.12	**96.19±0.05**
CIFAR-100	PreAct ResNet-18	**79.03**±0.30	78.1 ±0.60	78.75± 0.28
Tiny ImageNet		65.76±0.12	65.45±0.42	**66.16±0.50**

4 DISCUSSION

We have introduced region mixup, a simple extension of the mixup data augmentation principle. Integrating region mixup into existing mixup training pipelines requires just a few lines of code and adds minimal to no computational overhead. Through empirical findings, we observe the effectiveness of region mixup in visual recognition. We anticipate that Region Mixup will receive extensive investigation and further extensions, potentially becoming a valuable regularization tool for practitioners in deep learning.

ACKNOWLEDGEMENTS

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of the ICLR 2024 Tiny Papers Track.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. 2020.

Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Gradcam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, 2018. doi: 10.1109/WACV.2018.00097.

William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL `https://github.com/Lightning-AI/lightning`.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 3714–3722. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33013714. URL `https://doi.org/10.1609/aaai.v33i01.33013714`.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016. URL `https://api.semanticscholar.org/CorpusID:6447277`.

Hoki Kim. Torchattacks : A pytorch repository for adversarial attacks. *ArXiv*, abs/2010.01950, 2020. URL `https://api.semanticscholar.org/CorpusID:222132956`.

JangHyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=gvxJzw8kW4b`.

Zicheng Liu, Siyuan Li, Di Wu, Zhiyuan Chen, Lirong Wu, Jianzhu Guo, and Stan Z. Li. Automix: Unveiling the power of mixup for stronger classifiers. In *European Conference on Computer Vision*, 2021. URL `https://api.semanticscholar.org/CorpusID:238531789`.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, 2018. URL `https://api.semanticscholar.org/CorpusID:59604501`.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Young Joon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019. URL `https://api.semanticscholar.org/CorpusID:152282661`.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2015. URL `https://api.semanticscholar.org/CorpusID:6789015`.