Supplementary Material for Submission 6259

Anonymous submission

Mix-up Methods Summary

In this Section, we provide brief summary of some mix-up methods. We refer readers to the survey paper (Naveed 2021) for a more detailed overview.

- Input Mix-up (Zhang et al. 2018): first variant, simply interpolates two samples based on the weight sampled from the Beta distribution, then train the Deep Neural Network with the mixed samples.
- Manifold Mix-up (Verma et al. 2019) interpolates the two samples at a random layer in the DNN instead of the first layer.
- CutMix (Yun et al. 2019) selects a random rectangular region with size sampled from the Beta distribution from one image and pastes it and paste it to another.
- SaliencyMix (Uddin et al. 2021) works similar to CutMix, but it selects the top salient regions based on the saliency calculation.
- PuzzleMix (Kim, Choo, and Song 2020) first calculates the saliency of the image, then optimizes secondary objectives to ensure rich supervisory signal of the mixed image.
- Co-Mixup (Kim et al. 2021) extends PuzzleMix to the batch level instead of a pair of images by optimizing mixing objectives for the whole batch.
- AutoMix (Liu et al. 2022) separates mixing and classifying into separate parts and adds encoders to the training pipeline to automatically mix images and labels.
- AlignMix (Venkataramanan et al. 2022) calculates the distance of feature vectors, measures assignment matrix using Sinkhorn-Knopp algorithm, then mixes the images based on the matrix.

We provide comparisons of R-Mix with other mix-up methods in Table 1 and 2.

	Model	Eps	Cost/Eps	Accuracy
Vanilla	PARN-18	300	1/1	76.41
Input	PARN-18	300	1/1	77.57
Manifold	PARN-18	300	1/1	78.36
CutMix+	PARN-18	300	1/1	80.60
SaliencyMix	PARN-18	2000	2/1	80.31
PuzzleMix	PARN-18	300	2.9/1	79.38
Co-Mixup	PARN-18	300	3/1	80.13
AutoMix	RN-18	1200	2/1	80.95
AlignMix	PARN-18	2000	1.05/1	81.71
R-Mix	PARN-18	300	2/1	81.49

Table 1: Model, number of Epoch, Cost per Epoch and Accuracy of various mix-up methods on CIFAR-100. PARN: PreActResNet. RN: ResNet.

	Model	Eps	Cost/Eps	Accuracy
Vanilla	RN-50	100	1/1	75.97
Input	RN-50	100	1/1	77.03
Manifold	RN-50	100	1/1	76.70
CutMix	RN-50	100	1/1	77.08
SaliencyMix	RN-50	100	2/1	77.14
PuzzleMix	RN-50	100	2.9/1	77.51
Co-Mixup	RN-50	100	3/1	77.61
AutoMix	RN-50	100	2/1	77.91
AlignMix	RN-50	100	1.05/1	78.00
R-Mix	RN-50	100	2/1	77.41

Table 2: Model, number of Epoch, Cost per Epoch and Accuracy of various mix-up methods on ImageNet. PARN: PreActResNet. RN: ResNet. Note that at the time of writing, AlignMix has not released the 100 epochs training code nor the models for it.

Ablation Study: different ways to randomly mix images

In this section, we describe the design process that leads to the current implementation of R-Mix. We summarize the design steps in Table 3. Experiments are conducted on CIFAR-100 (Krizhevsky 2009) using PreActResNet-18 (He et al. 2016).

First, we apply the cut-and-paste strategy to the top salient regions of the image (analogy to SaliencyMix (Uddin et al. 2021), Strategy 1). The top salient region is randomly selected from the top-k value described in the main paper. We observe that it offers marginal improvement (0.5%) over PuzzleMix (Kim, Choo, and Song 2020).

Second, we apply cut-and-paste strategy to both the top and least salient regions, and keep the rest intact (Strategy 2). This strategy reduces the accuracy of the model by 0.36%.

Third, we apply mixing to the top-top and least-least salient regions, and select only the top patches in the top-least case (Strategy 3). This is the proposed R-Mix method. This strategy gives 81.49% accuracy, which is the best so far.

Finally, we apply the same strategy for the top-top and least-least regions, but select only the least region in the top-least case (Strategy 4). This time, the performance is significantly decreased by up to 6%.

Seeing that none of the strategy works as good as Strategy 3, we name it R-Mix.

Implementation details

We visualize the training pipeline of R-Mix in Figure 1. We will release our source code under MIT License upon acceptance. Here, we describe implementation details to reproduce the result:

- **CIFAR-100**. We use four model architectures: PreActResNet-18 (He et al. 2016), Wide ResNet 16-8 and 28-10 (Zagoruyko and Komodakis 2017), and ResNeXt 29-4-24 (Xie et al. 2016). Wide ResNet models do not use Dropout. All models are trained for 300 epochs, except WRN28-10 is trained for 400 epochs following the original implementation in PuzzleMix (Kim, Choo, and Song 2020). Augmentation includes Random Horizontal Clip and Random Crop with padding 2. Images are normalized channel-wise following well-known mean and standard deviation values. We train the models with SGD algorithm using batch size 100, Nesterov Momentum 0.9, and weight decay 0.0001. The OneCycleLR parameters are set as follows: div factor 100, final div factor 10000, max LR 0.3.
- **ImageNet**. For ImageNet (Russakovsky et al. 2015), we follow the 100 epoch training protocol used in Co-Mixup (Kim et al. 2021). We keep the training pipeline the same, replacing Co-Mixup part with R-Mix.

The code for R-Mix can be found at file `r_mix.py` file, in the codebase that is supplied with this document.

Sample visualizations

Sample visualizations are in Figure 2

References

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity Mappings in Deep Residual Networks. In *ECCV*.

Kim, J.; Choo, W.; Jeong, H.; and Song, H. O. 2021. Co-Mixup: Saliency Guided Joint Mixup with Supermodular Diversity. In *ICLR*.

Kim, J.-H.; Choo, W.; and Song, H. O. 2020. PuzzleMix: Exploiting Saliency and Local Statistics for Optimal Mixup. In *ICML*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report.

Liu, Z.; Li, S.; Wu, D.; Liu, Z.; Chen, Z.; Wu, L.; and Li, S. Z. 2022. AutoMix: Unveiling the Power of Mixup for Stronger Classifiers. *ECCV*.

Naveed, H. 2021. Survey: Image Mixing and Deleting for Data Augmentation. arXiv:2106.07085.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*.

Uddin, A. F. M. S.; Monira, M. S.; Shin, W.; Chung, T.; and Bae, S.-H. 2021. SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization. In *ICLR*.

Venkataramanan, S.; Kijak, E.; Amsaleg, L.; and Avrithis, Y. 2022. AlignMixup: Improving Representations by Interpolating Aligned Features. In *CVPR*.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In *ICML*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2016. Aggregated Residual Transformations for Deep Neural Networks. In *CVPR*.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *ICCV*.

Zagoruyko, S.; and Komodakis, N. 2017. Wide Residual Networks. arXiv:1605.07146.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *ICLR*.

		Usage	Mix strategy	Accuracy
Strategy 1 (SaliencyMix)	Top Salient	Yes	Cut-and-Paste	79.88%
	Least Salient	No	None	
Strategy 2	Top Salient	Yes	Cut-and-Paste	79.52%
	Least Salient	Yes	Cut-and-Paste	
Strategy 3 (R-Mix)	Top-Top Salient	Yes	Mix	**81.49%**
	Least-Least Salient	Yes	Mix	
	Top-Least Salient	Yes	Select Top only	
Strategy 4	Top-Top Salient	Yes	Mix	75.42%
	Least-Least Salient	Yes	Mix	
	Top-Least Salient	Yes	Select Least only	

Table 3: Design steps that lead to the implementation of R-Mix. We try different ways to mix images, and then observe Strategy 3 offers the best result. We study on CIFAR-100 using PreActResNet-18.

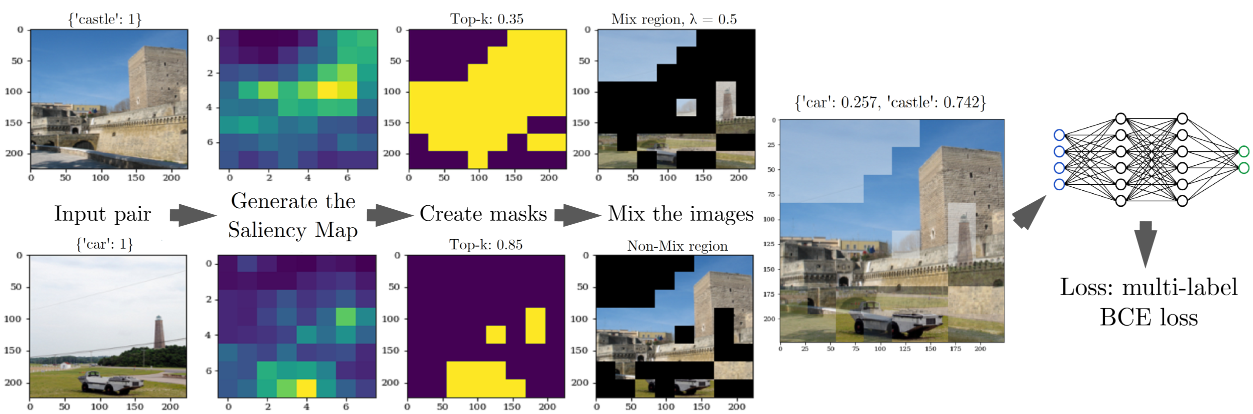

Figure 1: Training pipeline of R-Mix. First, it calculates the saliency map, then divies the map into two regions. Next, it mixes the images based on the region the patch belongs to. Finally, it combines the number of patches and mixing ratio to determine the weights of the inputs.

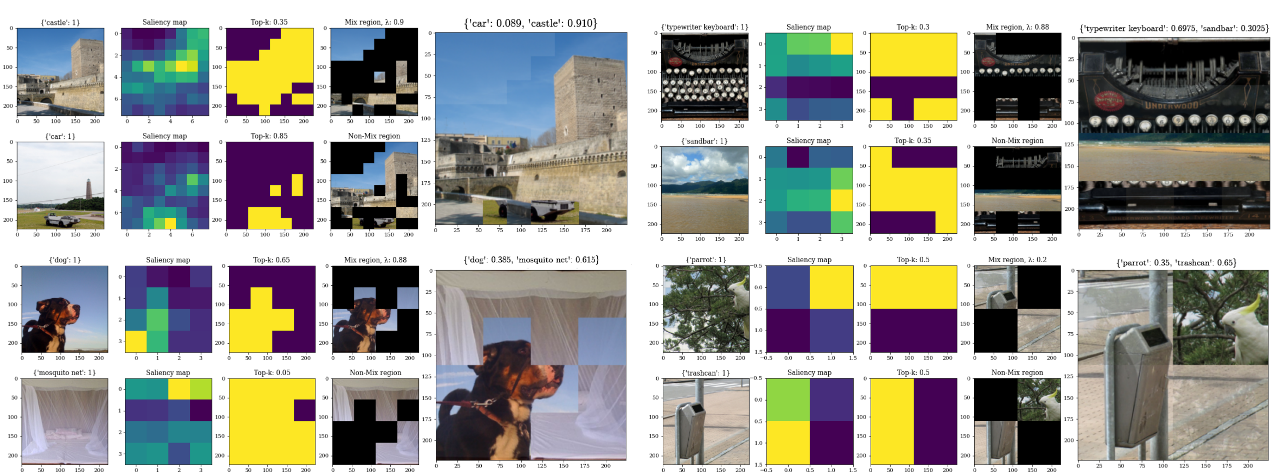

Figure 2: Sample visualization of images produced by R-Mix.