Modern data augmentation, why so unintuitive?

Author: **Dominik Lewy**[1], Lingaro Group

Notation

Let us start with an introduction of the base notation. First of all, mixing DA techniques are divided into two main classes: those that mix images using pixel-wise weighted average (referred to as pixel-wise mixing) and those that mix images spatially by means of extracting patches from different images and joining them together (referred to as patch-wise mixing methods). Examples from both classes are presented in Figure 1.

Figure 1. From left to right: two sample images and examples of pixel-wise and patch-wise mixing, respectively. The pixel-wise and patch-wise images present the zoomed region indicated by a red rectangle to show the detailed characteristics of the mixed images.

Augmentation by mixing images - methods map

Figure 2 presents a map of the mixing methods, indicating for each of them the publication date, certain key characteristics and relations to other methods.

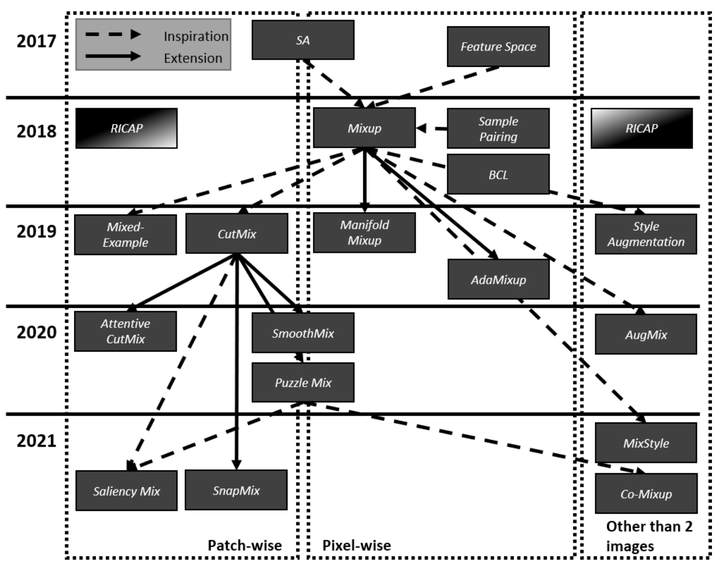

Figure 2. Image mixing DA methods presented on a time scale, with key characteristics and dependencies indicated. Dotted regions separate methods in which mixing takes pixel-wise form (Pixel-wise) from those with spatial mixing (Patch-wise), and those with mixing applied not to a pair of images, but either just one image and its transformed version or more than 2 images (Other than 2 images). Directed lines indicate inspirations (dotted lines) or direct extensions (solid lines) of the methods.

(Almost) Universal image mixing equation

Image mixing DA methods rely on blending two input images and their corresponding labels according to the following equations:

$$\tilde{x} = B \odot x_1 + (I - B) \odot x_2 \quad (1)$$

$$\tilde{y} = \lambda y_1 + (1 - \lambda) y_2 \quad (2)$$

where x_1, x_2 are original input images, y_1, y_2 are one-hot label encodings, λ is a mixing ratio, B is a mixing mask matrix suitable for both pixel-wise and patch-wise mixing and I is an identity matrix of the same dimensionality as B. $\odot$ denotes element-wise matrix multiplication operation. The vast majority of approaches described in this section are built around equations (1)-(2) and mainly differ by the method of λ selection and construction of matrix B.

Canonical method - Mixup

A founding mixing method is Mixup introduced in 2018. Mixup constructs new training samples according to equations (1)-(2) by means of the same weighted mean of the images and their labels, i.e. the entire matrix B is populated with λ. The underlying assumption of Mixup is that linear interpolation of feature vectors should lead to an adequate linear combination of the associated labels. This linear combination of images / classes is controlled by λ, e.g., λ = 0.5 leads to averaging the images and their corresponding labels, while $\lambda \in \{0, 1\}$ preserves one of the original images and its label. The effects of Mixup is presented in Figure 3.

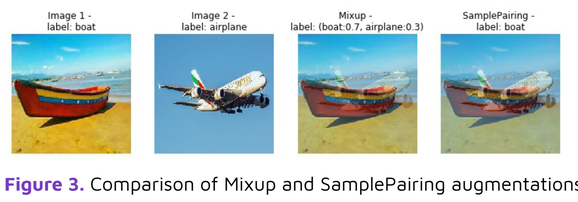

Figure 3. Comparison of Mixup and SamplePairing augmentations.

Canonical method - CutMix

The other stream of Mixup follow-up papers, which question the efficacy of mixing pixels linearly, share many properties with the augmentation methods focused on occluding parts of the image. An underlying assumption of this stream of methods is that linear interpolations represent just a small subset of mixing operations that can potentially be used for data augmentation. An approach worth mentioning here is CutMix method which is illustrated in Figure 4.

Empirical evaluation on CIFAR-10

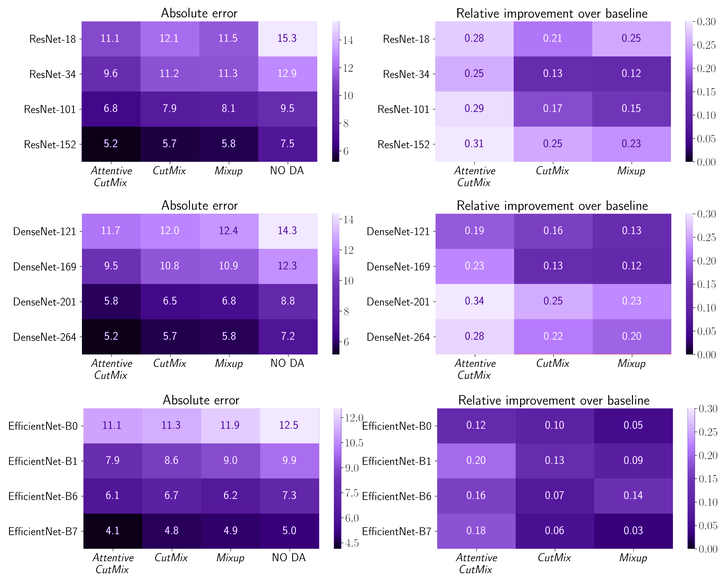

Figure 5. Accuracy results for Mixup, CutMix and AttentiveCutMix on CIFAR-10. Each panel presents a particular type of architecture (ResNet, DenseNet, EfficientNet) with complexity of the network increasing from top to bottom. Left panels present an absolute error value and the right ones a relative improvement over the baseline ("NO DA" results).

Looking at the left panels of the figure 5, one can conclude that DA improves the accuracy regardless of particular architecture type and complexity. In all three left panels the error decreases when gradually more complex architectures are used (top to bottom) and with an application of more advanced DA methods (right to left). It is generally assumed that Mixup, as the initial method in the area, is the least advanced, followed by CutMix, developed based on a certain criticism of Mixup, and AttentiveCutMix which is an extension of CutMix.

Another way to look at the results is from the perspective of a relative improvement over the baseline (the right panels). A general observation is that using DA with more complex types of architectures yields lower relative boost, on average equal to around 22%, 21% and 11%, respectively for ResNet, DenseNet and EfficientNet. However, the relative effects of DA vary substantially within each architecture type. The highest relative advantage is achieved by the most complex model (ResNet-152) in ResNet group, but for the other two architectures the highest boost is observed for DenseNet-201 and EfficientNet-B1, respectively, which are not the most complex ones.

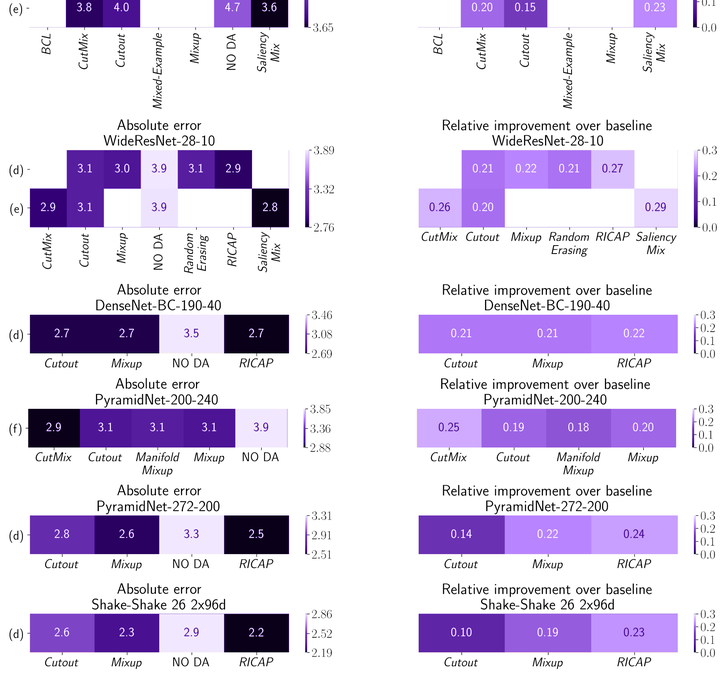

Figure 6. Accuracy results of various augmentation methods on CIFAR-10 grouped around common baselines (particular architectures). Left panels present absolute error values and the right ones the relative improvements over the baseline.

Figure 6 shows the results of experiments grouped around the same baselines (i.e. particular architectures) for a wider selection of augmentation methods. Generally, similar trends can be observed as in the case of groups of architectures. Simpler models benefit relatively more from data augmentation, however, in terms of absolute figures they still yield higher errors. All in all, every DA technique is able to improve over the baseline, with patch-wise methods achieving slightly better results. For both ResNet-18 architectures best result were accomplished by patch-wise (Saliency Mix) method, for PyramidNet-200-240 by CutMix (a patch-wise mixing approach), and for the remaining architectures RICAP, which is also a patch-wise mixing method, performed better or equally good than its competitors. The results additionally confirm that erasing methods (Cutout and RandomErasing) are slightly inferior to the mixing ones.

Architecture	Method	Error
Shake-Shake 26 2x96d	RICAP	2.19
Shake-Shake Regularization	BC+	2.26
Shake-Shake 26 2x96d	Mixup	2.32
PyramidNet-272-200	RICAP	2.51
PreAct ResNet-34	Manifold Mixup	2.54

Table 1. Top-5 overall best outcomes on CIFAR-10.

The last comparison, presented in Table 1, lists top-5 combinations of an architecture and an augmentation method for CIFAR-10 found in the literature. The list is led by two versions of the Shake-Shake model.

Similarities of the methods

Method	Where		How			Pixel-wise			Mix labels		Loss function		Nr. images		Computational complexity
	I	H	AM	O	R	Yes	No	Both	Yes	No	CEE	Non-CEE	3+	1	2	
AdaMixup	•		•			•			•			•		•		B
Attentive CutMix	•		•				•		•		•			•		B
AugMix	•			•		•				•		•	•			B
BCL	•			•		•			•			•		•		A
Co-Mixup	•		•					•	•		•		•			B
CutMix	•			•			•		•		•			•		A
Feature Space		•		•		•				•	•			•		B
Manifold Mixup		•		•		•			•		•			•		B
Mixed-Example	•			•			•		•		•			•		A
MixStyle		•		•		•				•	•			•		B
Mixup	•			•		•			•		•			•		A
Puzzle Mix	•		•					•	•		•			•		B
RICAP	•			•			•		•		•		•			A
Saliency Mix	•		•				•		•		•			•		B
Sample Pairing	•			•		•				•	•			•		B
Smart Augmentation	•		•					•		•		•		•		B
SmoothMix	•			•			•		•		•			•		A
SnapMix	•			•		•			•		•			•		B
Style Augmentation	•		•			•				•	•		•			B

Table 2. Comparison of data augmentation techniques with respect to particular baseline properties: where, how and in which form the augmentation is applied, whether or not it mixes labels or utilizes a specific loss function, how many images take part in a single augmentation and what is the computational complexity of the method. I - input layer, H - hidden layer, AM - auxiliary mechanism (either network or other), O - optimization, R - rule, CEE - Categorical Cross Entropy, A - there is no significant computational overhead, B - requires either special training process, multiple evaluations or an auxiliary component that incurs additional computational cost.

Application to image-related tasks other than classification

The canonical tasks in Computer Vision are image categorization, object localization, object detection and semantic segmentation. When it comes to application of certain data augmentation techniques to various tasks, there are two key properties: (1) whether the mixing is performed pixel-wise or patch-wise and (2) how many images are mixed. Consequently the methods could be divided into 3 following groups:

- Group A - Pixel-wise or mixed Pixel-wise and Patch-wise augmentations that work on 2 or more images. Those methods are limited to categorization task due to their underlying property of mixing images-pixel wise. This leads to certain regions of the image representing more than one class and renders application of these methods to other tasks difficult (e.g. what should be done with a bounding box for a part of the image that is mixed?).
- Group B - Patch-wise methods directly address categorization and localization tasks and can be further adjusted to object detection and segmentation by proper handling of an additional information associated to the task (e.g. by limiting the bounding box to the area corresponding to the selected patch).
- Group C - methods that work on just 1 image, can in principle be applied to all tasks.

There are also certain methods that were developed with a particular problem in mind, e.g. SnapMix. The method is dedicated to fine-grained image classification problem, in which classes are differentiated by details only.

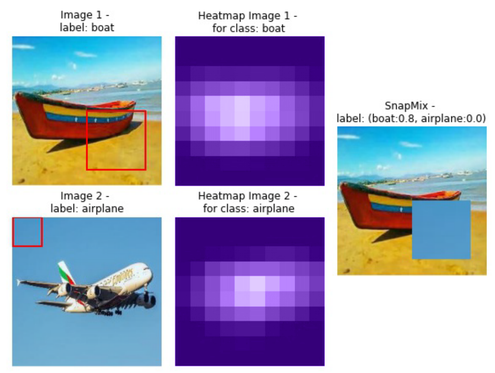

Figure 7. Effects of SnapMix augmentation with the corresponding mixed label. Left: original images with randomly selected patches of different sizes. Middle: heatmaps presenting the output of CAM for the respective class. Right: a resulting image after SnapMix application. Observe that elements of the label vector (right figure) do not have to sum up to 1.

In which modalities can we apply those methods

When it comes to applying image augmentation methods to other modalities the following 3 groups could be distinguished:

- Group A - Mixup-like methods (Pixel-wise mixing) that work on 2 or more images and do not utilize any complex mixing mechanism. Those methods can be applied to other modalities without any adaptation as long as the same size of the input objects is ensured. In the context of audio it means having the same length and the same spectrum of frequencies, and for text data, the same size of vector embeddings.
- Group B - Patch-wise methods or mixed Pixel-wise and Patch-wise. Their application to modalities other than images is technically possible, however, not yet empirically tested. Such a mixing would potentially signify specific modality-depending aspects, e.g. spatial mixing of embeddings of different sentences or pasting a part of a voice spectrogram into another one.
- Group C - methods that cannot be directly applied to other modalities due to their inherent connection to image-specific data transformations or architectures. A potential application of the methods from group C to other modalities would require introducing major changes to their design and operation, as they are inherently related to image data. Some methods from this group utilize image saliency information, other use image specific data transformations, like style transfer or rotation. Yet another ones, utilize architectures dedicated to processing the image data.

References

References for all methods referred to in the poster using italic as well as for all the empirical results can be found in the below survey paper.

[1] Lewy D. Mańdziuk J. An overview of mixing augmentation methods and augmentation strategies. Artif Intell Rev, 2022.