# Improving Data Augmentation for Medical Image Segmentation

**Zach Eaton-Rosen**
CMIC, UCL
z.eaton-rosen@ucl.ac.uk

**Felix Bragman**
CMIC, UCL

**Sebastien Ourselin**
BMEIS, KCL

**M. Jorge Cardoso**
BMEIS, KCL
CMIC, UCL

## Abstract

Medical image segmentation is often constrained by the availability of labelled training data. ‘Data augmentation’ helps to prevent memorisation of training data and helps the network’s performance on data from outside the training set. As such, it is vital in building robust deep learning pipelines. Augmentation in medical imaging typically involves applying small transformations to images during training to create variety. However, it is also possible to use linear combinations of training images and labels to augment the dataset using the recently-proposed ‘mixup’ algorithm. Here, we apply this algorithm for use in medical imaging segmentation. We show that it increases performance in segmentation tasks, and also offer a theoretical suggestion for the efficacy of this technique.

## 1 Introduction

‘Data augmentation’ is used to artificially increase the size of the training dataset. In medical imaging, this is typically done with transformations that are applied to both the images and labels equally, creating warped versions of the training data. Augmentation methods commonly employ transformations such as rotations, reflections, and elastic deformations, which produce training images that closely resemble one particular training example. While the intuitive motivation behind augmentation strategies is appealing, a recently-proposed technique, ‘mixup’ [8] works by training on linear combinations of existing training data: the training labels are also linear combinations of the ground-truth labels. Although images generated in this manner are noticeably different than training images (looking like two images super-posed), this augmentation technique has been shown to improve performance on a variety of machine-learning tasks.

In this work, we apply mixup to medical image data for the purpose of semantic segmentation. We also introduce a variant on the technique, which we label ‘mixmatch’. This variant takes class prevalence into account when deciding which patches to mix. Both mixup and mixmatch improve segmentation results on the BraTS dataset. Code for this work will be made available via the open-source medical imaging software NiftyNet [2].

## 2 Methods

In mixup [8], images from the training set are combined such that a mixup sample is a linear combination of two training data: $(x_i, y_i)$ and $(x_j, y_j)$. The mixup sample is given by: $(x_{mixup} = \lambda x_i + (1 - \lambda)x_j, y_{mixup} = \lambda y_i + (1 - \lambda)y_j)$. The parameter $\lambda \in [0, 1]$ and is distributed according to a Beta distribution: $\lambda \sim \beta(\alpha, \alpha)$ for $\alpha \in (0, \infty)$. In mixup, the samples to be combined are chosen randomly from all available images (in our case, patches). Here, we use mixup and also a proposed we call ‘mixmatch’, in which the mixing is not totally random.

1st Conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam, The Netherlands.

The motivation for mixmatch comes from noting that datasets in medical imaging are often highly imbalanced. For instance, in the BraTS dataset, the majority of voxels in the image contain no tumour. We thus propose a simple alteration: instead of $x_{i,j}$ being chosen at random from all the loaded image patches, we instead match the patches with highest foreground amounts with the lowest. Implementation-wise, for each minibatch of size $n$ that we use, we load $2n$ patches. From the first set of $n$ patches, we select the highest concentration of foreground and match it with the lowest concentration from the $2^{nd}$ $n$ patches.

### 2.1 Performance on BraTS training set

Mixup has been used on whole-image classification problems, but not semantic segmentation. Firstly, we test whether mixup benefits training in semantic segmentation. We then test our proposed variant, mixmatch. For data, we use the BraTS 2017 dataset [1, 4] — a multi-modal MRI dataset of labelled brain gliomas. We used the network architecture of the 2nd-placed entry in BraTS 2017: A cascaded neural network [7] (the winning entry was an ensemble of networks rather than a single network[3], which would have increased the training burden). We trained the network to predict the label ‘whole tumour’ binary label and trained using the Dice loss [5], fixing the mixup $\alpha = 0.4$ in all experiments.

Good augmentation should increase the generalisability of the learned network. To test the effects of this, we independently trained the network on a small fraction of the BraTS dataset (10/285 subjects chosen at random).

In total we trained 4 networks on large (199 subjects) and small (10) subsets of the BraTS dataset. In labelling these results, ‘no mix’ refers to no augmentation; ‘aug’ refers to augmenting with rotations, random flips (left-right) and zoom; ‘mixup’ is vanilla mixup and ‘mixmatch’ has patches being matched depending on the amount of foreground label in the image.

### 2.2 Effect of mixup/mixmatch samples on training

Training neural networks relies on the backpropagation algorithm. By using the chain rule, the effect of a given pixel’s value, $p_i$, on the final loss function is propagated to update the network’s parameters. We wanted to investigate whether $\frac{\partial \mathcal{L}}{\partial p_i}(\lambda x_i + (1-\lambda)x_j) \approx \lambda\frac{\partial \mathcal{L}}{\partial p_i}(x_i) + (1-\lambda)\frac{\partial \mathcal{L}}{\partial p_i}(x_j)$. If this holds, it may suggest that mixing samples acts somewhat like increasing the batch-size. To perform a first check of this idea, we combined two images with varying $\lambda$ to observe the effects on $\frac{\partial \mathcal{L}}{\partial p_i}$. We calculate this derivative using a trained network from 2.1.

## 3 Results

In Figure 1, we see the performance of the various methods, plotted against the iteration. Mixup and mixmatch both outperform both un-augmented training and augmented training at every iteration. Mixmatch has similar performance to mixup.

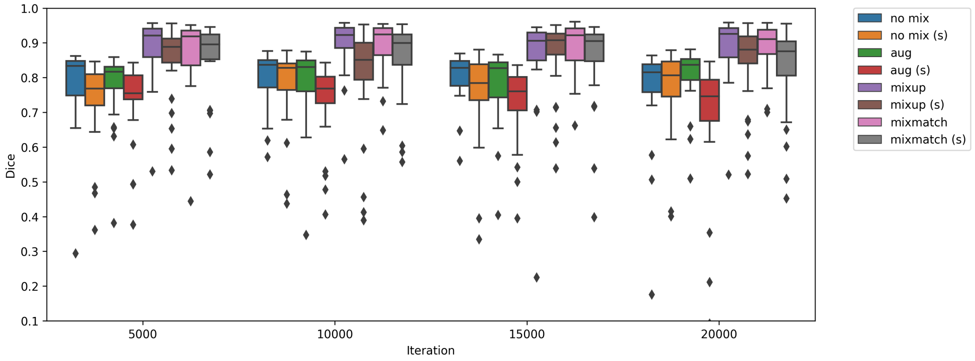

Figure 1: Results on withheld subjects for Section 2.1. For each of the plotted boxes, the performance is measured on the same random subjects from outside of the training set. The results from the small training set are denoted by ‘(s)’. Mixup and mixmatch improve the Dice scores over the alternatives.

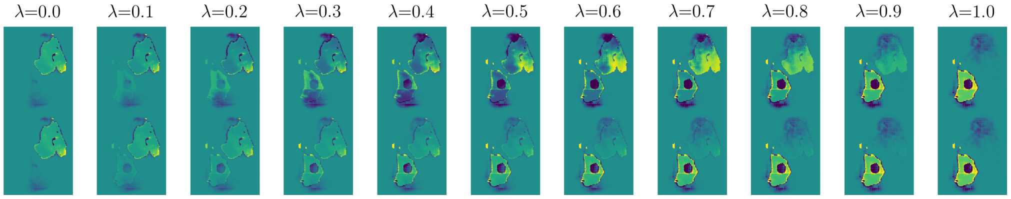

Figure 2: Left-Right: $\lambda$ is varied between 0 and 1 to make a mixed image. Top row: the computed $\frac{\partial\mathcal{L}(\lambda x_i+(1-\lambda)x_j)}{\partial p_i}$. Bottom row: the weighted average $\lambda\frac{\partial\mathcal{L}(x_i)}{\partial p_i}+(1-\lambda)\frac{\partial\mathcal{L}(x_j)}{\partial p_i}$ of the unmixed images. The top and bottom rows are similar, especially at either end of this spectrum.

In terms of the effects of mixup/mixmatch on $\frac{\partial\mathcal{L}}{\partial p_i}$, see Figure 2. We observe that especially for small mixing fractions, the values of $\frac{\partial\mathcal{L}}{\partial p_i}$ are noticeably similar to the weighted averages of their independent loss derivatives. This may suggest that some of the benefits of mixup could be seen as analogous to mini-batch training: the dice loss derivatives are close to a weighted sum of the derivatives from mixed patches. At $\lambda = 0.5$, this would be comparable to a mini-batch of size 2, although more experiments would be required to further validate this suggestion.

## 4 Discussion

Mixup is an augmentation technique with low computational overhead that improves semantic segmentation results. These early results suggest performance increases with little overhead. Although the theoretical justification for this non-intuitive augmentation technique is not settled, we believe that Figure 2 sheds some light on its success by relating it to mini-batch training. This benefit could be especially relevant in medical imaging, where our large images mean batch-sizes are far below their typical values in other computer-vision settings. We also note that mixup acts to smooth the labels, which is a regularisation technique in its own right [6]. In future, we will investigate whether variants of mixup (including the proposed mixmatch) may help to address issues such as class imbalance which are prevalent in medical imaging studies. To this end, we note that mixmatch may benefit from insight from the curriculum-learning literature in terms of which patches should be matched.

**Acknowledgements:** Zach Eaton-Rosen is supported by the EPSRC Doctoral Prize. We gratefully acknowledge the support of NVIDIA Corporation for their donation of the GPUs used in this research.

## References

[1] S. Bakas et al. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117, 2017.

[2] E. Gibson et al. Niftynet: a deep-learning platform for medical imaging. *Computer Methods and Programs in Biomedicine*, 158:113 – 122, 2018.

[3] K. Kamnitsas et al. Ensembles of multiple models and architectures for robust brain tumour segmentation. *arXiv preprint arXiv:1711.01468*, 2017.

[4] B. Menze et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2015.

[5] F. Milletari et al. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[6] C. Szegedy et al. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[7] G. Wang et al. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. *CoRR*, abs/1709.00382, 2017.

[8] H. Zhang et al. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.