MixNet: Multi-modality Mix Network for Brain Segmentation

Long Chen and Dorit Merhof(✉)

Institute of Imaging and Computer Vision, RWTH Aachen University, Aachen, Germany
{long.chen,dorit.merhof}@lfb.rwth-aachen.de
https://www.lfb.rwth-aachen.de/

Abstract. Automated brain structure segmentation is important to many clinical quantitative analysis and diagnoses. In this work, we introduce MixNet, a 2D semantic-wise deep convolutional neural network to segment brain structure in multi-modality MRI images. The network is composed of our modified deep residual learning units. In the unit, we replace the traditional convolution layer with the dilated convolutional layer, which avoids the use of pooling layers and deconvolutional layers, reducing the number of network parameters. Final predictions are made by aggregating information from multiple scales and modalities. A pyramid pooling module is used to capture spatial information of the anatomical structures at the output end. In addition, we test three architectures (MixNetv1, MixNetv2 and MixNetv3) which fuse the modalities differently to see the effect on the results. Our network achieves the state-of-the-art performance. MixNetv2 was submitted to the MRBrainS challenge at MICCAI 2018 and won the 3rd place in the 3-label task. On the MRBrainS2018 dataset, which includes subjects with a variety of pathologies, the overall DSC (Dice Coefficient) of 84.7% (gray matter), 87.3% (white matter) and 83.4% (cerebrospinal fluid) were obtained with only 7 subjects as training data.

Keywords: Brain segmentation · CNN · Multi-modality

1 Introduction

Accurate automated segmentation of brain structures, e.g., white matter (WM), gray matter (GM), and the cerebrospinal fluid (CSF) forms the basis for high-throughput quantitative analyses and associated diagnoses. while computed tomography (CT) and positron emission tomography (PET) is also used for brain structure analysis, magnetic resonance imaging (MRI) is the most popular choice [1]. We will only talk about MRI in this work.

As the deep learning approaches are becoming mature, they gradually outperforms previous methods [2–5]. Based on the network architecture, these deep learning approaches can be roughly divided into two categories: the patch-wise [6–8] and semantic-wise [9] architecture. The patch-wise approach takes

A. Crimi et al. (Eds.): BrainLes 2018, LNCS 11383, pp. 367–377, 2019.
https://doi.org/10.1007/978-3-030-11723-8_37

a local patch around a pixel as input. Most of the current works use this strategy, because of its efficiency of using the training dataset. Compared to the semantic-wise approach, the patch-wise approach can extract large number of patches from the MRI subjects for training. But unlike unstructured segmentation, brain structures preserve same relative positions in all subjects and patch-wise approaches ignores that information. Some works like [8] make up for this by augmenting the network input with coordinates of voxels, but semantic-wise methods still have advantages in nature.

In addition to the overall architecture, we can also use input dimensions to distinguish between different methods. The 3D networks leverage the spatial information more efficiently than 2D networks by sharing kernels across three dimensions. The cost is longer runtime and limited network size. As discussed in Sect. 2.3, the 2D network can observe the 3D MRI volume from different directions, that is, more 2D slices as training data. This strategy does not only provides more training images but also plays the role of a ensemble model. By fusing the results obtained from 2D slices along different orientations the segmentation should be more robust and spatially consistent as well.

We propose a 2D semantic-wise CNN to handle the brain structure segmentation problem in Sect. 2. Three structures are tested to see the effect of different ways of mixing multiple modalities. We call them MixNetv1, MixNetv2 and MixNetv3 in Sect. 2.2. The experiments are performed with the MICCAI challenge MRBrainS2018 dataset. The dataset contains annotated multi-sequence (T1-weighted, T1-weighted inversion recovery and T2-FLAIR) scans of 30 subjects. Seven of them are distributed as training data, while the rest subjects are kept unreleased for test. For a limited training dataset, the transfer learning [10] usually boosts the overall segmentation results. But this is achieved by using extra data implicitly. Our experiment works with only 7 subjects of the MRBrainS2018 training dataset.

The code developed for this work and trained models will be available online: https://github.com/looooongChen/MRBrainS-Brain-Segmentation.

2 Method

In Sect. 2.1 we introduce the residual dilated convolution unit. Except the initial convolution layer and the output module, MixNet is composed of residual dilated convolution units connected in series or parallel. Section 2.2 discusses different ways of using multi-modalities. Section 2.3 describes the method of acquiring more 2D training slices from the 3D MRI volume.

2.1 Basic Units of the Nets

As shown in Figs. 4, 5 and 6, the networks are composed of three types of basic units: the InitUnit (Fig. 1), the DilateResUnit (Fig. 2) and the OutputUnit (Fig. 3). In this section, we will described them in detail.

Initial Unit (InitUnit). The InitUnit consists of a single 5×5 convolutional layer and an optional pooling layer. Depending on the input channels, the convolution kernels can be of different sizes. In Fig. 4, three modalities are stacked together, while mixNetv2 (Fig. 5) and mixNetv3 (Fig. 6) have three input streams. Thus, the kernel sizes are $5 \times 5 \times 3$ and $5 \times 5 \times 1$, respectively. In addition, the pooling layer aims to reduce memory usage when necessary. If the pooling layer in the InitUnit is used, the upscaling layer in the OutputUnit should also be activated. In this work, we use a 2×2 pooling with stride 2.

Residual Dilated Convolution Unit (DilateResUnit). The training difficulty varies in different network architectures. For example, the degradation phenomenon arises in practice for a deeper plain CNN, although it includes the solution space of a shallower one. [11] conjecture that the deep plain CNN may have exponentially slow convergence rates and provides empirical evidence showing that a network composed of residual units is easier to optimize. The proceeding work [12] argues that the training procedure benefits from a "direct" path for information propagation, not only within a residual unit but through the whole network. Inspired by the successful works [11,12], we construct a deep residual learning network (DilateResUnit) with 'clear' paths through the layers and multiple modality streams for information propagation.

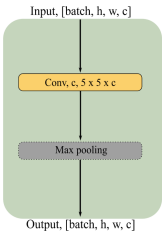


Fig. 1. InitUnit

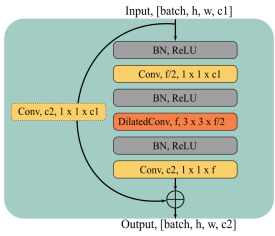


Fig. 2. DilateResUnit

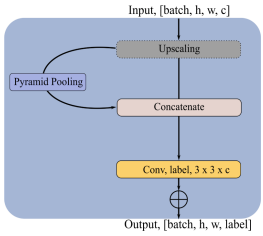


Fig. 3. OutputUnit

As shown in Fig. 2, the shortcut lets the input feature map pass through the unit directly and only the differences between inputs and outputs are learned. When such units are connected to form a network, these short paths will also be interlinked throughout the network. Compared to the residual unit in [12], the second convolutional layer is replaced by a dilated convolutional layer. Alternating convolutional layers and polling layers are a CNN common structure. The Pooling layer increases the receptive field efficiently while keeping the computational workload reasonable. However, the pooling layer loses localization information which is critical for segmentation tasks. Deconvolutions [13] and dilated convolutions (also known as atrous convolution) [14] are possible solutions. Different from the deconvolution where extra layers are involved to recover lost resolution, the dilated convolution keeps the resolution unchanged through the forward propagation. Extra layers mean more parameters. Assuming a network with less parameters is easier to train, we adopt the dilated convolution in this work.

A DilateResUnit is determined by four parameters: $c1, c2, f$ and d. The number of filters and the filter size of the dilated convolutional layer is $f/2$ and f, while d is the dilation factor. The first and last 1×1 convolutional layers are determined by the channels of the input and output feature map. When the inputs and outputs are of different sizes, a 1×1 convolutional layer will be inserted on the shortcut. Since we use the same f through the network, only the units before and after a concatenation in Fig. 5 (except the final concatenation) have such shortcut convolutions.

Output Unit (OutputUnit). As discussed in Sect. 1, anatomical structures preserve certain relative positions. Thus, the OutputUnit augments the input feature map with a global prior first, and then outputs results through a 3×3 convolutional layer. The global prior is captured by a pyramid pooling module [15]. The pyramid pooling module separates the input feature map into subregions and forms representation by average pooling. Then, bilinear interpolation is performed to get the same size as the original feature map. In this work, we use a four-level pyramid with 2×2, 4×4, 6×6, 12×12 bins respectively.

Finally, the upscaling is performed to recover the original resolution, only when the pooling layer in the InitUnit is used. If the network can fit into the memory, pooling and upscaling are not necessary.

2.2 Network Architecture

In this section, we discuss three styles of using multiple modalities: stacked channels, periodic summarization and parallel streams. Correspondingly, three network architectures (MixNetv1, MixNetv2 and MixNetv3) are constructed with the units introduced in Sect. 2.1 to test the effect on the results.

At the output end, all three networks aggregate features form different levels. A multi-modality, mutli-scale feature map is then passed to the OutputUnit. which augments the feature map with a global prior and makes the final prediction. Detailed network parameters are listed in Table 1.

To train the network, we compute the cross-entropy loss of each pixel in an image and accumulate them as the training loss. In this work, all pixels are treated equally, ingoring the label imbalance. The training process can run streadily in this way, but labels of a relatively small number may not receive enough attention. Weighing pixels of different labels is an approach worth trying.

Stacked Channels (MixNetv1). A straightforward way to fuse multiple modalities is to stack them as different channels. Thus, the input of MixNetv1 is a batch of 3-channel images. The forward propagation path is composed of serially connected DilateResUnits. Since the output of a DilateResUnit has a similar resolution with the input, we set the the filter number of all units to the same. In this way, the feature map size and the corresponding computation are balanced throughout different layers.

Periodic Summarization (MixNetv2). MixNetv2 is a network architecture between MixNetv1 and MixNetv3. MixNetv1 fuses the multiple modalities at the very beginning, while MixNetv3 keeps different modality streams independent until the final output. In MixNetv2, periodic summarization of multi-modality information is performed. As shown in Fig. 5, Level 1, Level 3 and Level 5 play such a role. The summarization is then fed back to each modality stream.

Parallel Streams (MixNetv3). Three modality streams propagate forward independently in MixNetv3. Features from three streams are only collected when the OutputUnit makes the final prediction. Actually, the solution space of MixNetv3 is contained in MixNetv1. Each neuron in MixNetv1 has connection to all three modalities (indirect connections considered). If we force each neuron to connect to only one modality by setting some network parameters to 0, MixNetv1 can be equivalent to MixNetv3. However, MixNetv3 performs better than MixNetv1 based on our experiments. Experiment results are demonstrated in Sect. 3.

Table 1. Parameters of three MixNet versions. The input channel $c1$, filter number f, dilation factor d and output channel $c2$ of the DilateResUnit are listed with respect to the network level. As described in Sect. 2.1, the DilatedResUnit is fully determined by these four parameters.

MixNetv1	Level 1	Level 2	Level 3	Level 4	Level 5
Input	$120 \times 120 \times 72$	$120 \times 120 \times 72$	$120 \times 120 \times 72$	$120 \times 120 \times 72$	$120 \times 120 \times 72$
Filters	72	72	72	72	72
Dilation	2	1	4	1	8
Output	$120 \times 120 \times 72$	$120 \times 120 \times 72$	$120 \times 120 \times 72$	$120 \times 120 \times 72$	$120 \times 120 \times 72$
MixNetv2	Level 1	Level 2	Level 3	Level 4	Level 5
Input	$120 \times 120 \times 72$	$120 \times 120 \times 48$	$120 \times 120 \times 72$	$120 \times 120 \times 48$	$120 \times 120 \times 72$
Filters	24	24	24	24	24
Dilation	2	1	4	1	8
Output	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$
MixNetv3	Level 1	Level 2	Level 3	Level 4	Level 5
Input	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$
Filters	24	24	24	24	24
Dilation	2	1	4	1	8
Output	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$	$120 \times 120 \times 24$

2.3 View MRI Volume from Different Directions

For a 2D CNN, the 3D MRI volume can be observed from any direction. The most commonly used are the three anatomical planes: the sagittal plane, the

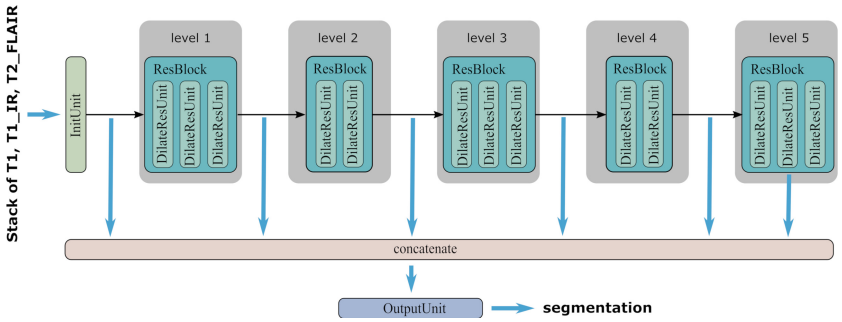


Fig. 4. MixNetv1: multiple modalities are stacked at the very beginning.

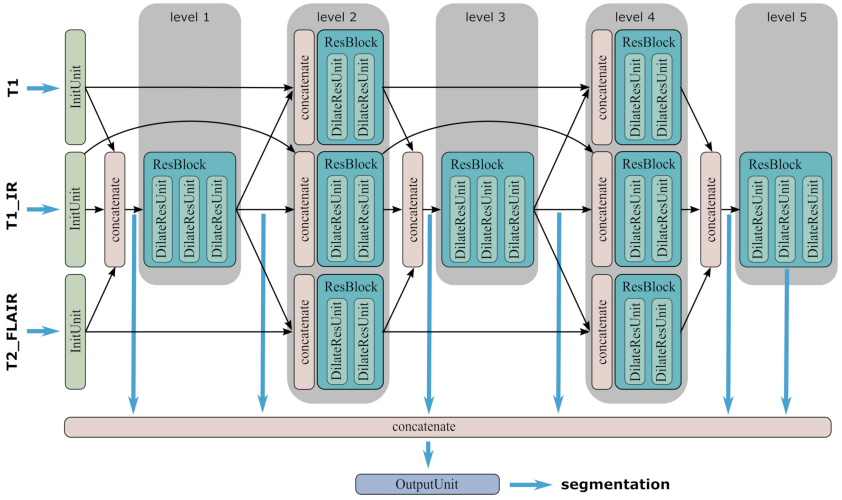


Fig. 5. MixNetv2: summarization of multi-modality information is performed periodically, then the summarization is fed back to each modality stream.

coronal plane and the transverse plane. By viewing the MRI volume from different directions, multiple batches of 2D slices can be acquired for training. For example, a $120 \times 120 \times 120$ volume will generate 360 images of the three anatomical planes. In fact, more directions can be included.

On one hand, changing the observation direction provides more training images. On the other hand, fusing predictions is actually an ensemble model, which improves the algorithm robustness and benefit the spatial consistency.

The annotation resolution of the MRBrainS2018 dataset is anisotropic in three directions. Therefore, this strategy cannot be fully utilized. We train three networks on the sagittal, coronal, transverse plane and fuse the predictions. Further tests can be performed by training a single classifier with images acquired along different orientations.

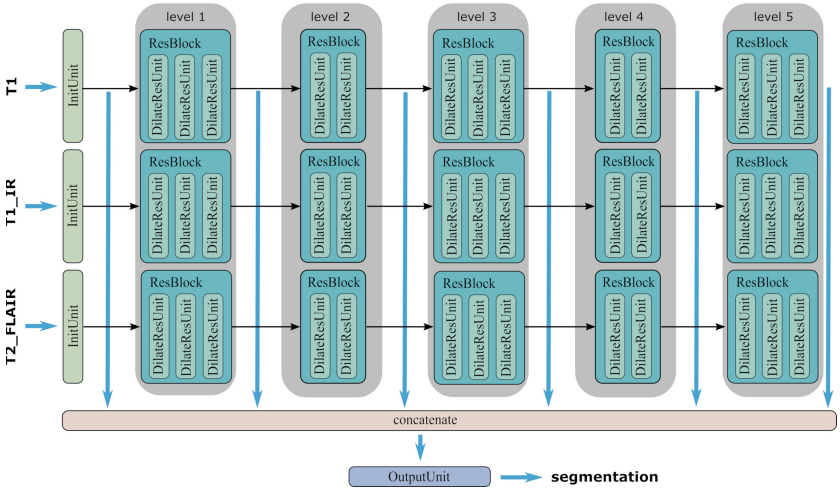


Fig. 6. MixNetv3: modality streams are kept separate until the OutputUnit aggregates information from each stream.

3 Results

The experiments are performed with the MICCAI challenge MRBrainS2018 dataset. The challenge releases 7 MRI scans (including T1-weighted, T1-weighted inversion recovery and T2-FLAIR) as the training data. Another 23 scans are kept secret for test. We test the three networks using leave-one-out cross validation strategy with the training dataset. MixNetv2 is submitted to the challenge and an evaluation of MixNetv2 on the test dataset is performed by the challenge organizers.

3.1 Preprocessing and Data Augmentation

Bias field correction [16] and image registration are performed by the challenge organizer. In addition to this, we linearly scale each modality image of each scan to have zero mean and unit variance.

To train the very deep network, the data is heavily augmented with elastic deformation [17], scaling, rotation and translation. As for the sagittal and coronal plane, the resolution in horizontal and vertical directions are four times different. Thus, we only apply flipping and translation.

It is worth mention that excessive elastic deformation and scaling may lead to an unstable training. We use scaling factors of 0.9, 0.95, 1.05 and 1.1, elastic deformation factor $\alpha = 10$ and $\sigma = 4$ [17] in this work. Rotation is performed around the image center with 8 degrees: 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°. The random translation is limited to 0.15% of the image size. We use all augmentation methods separately, that is, no images are generated from augmented images.

3.2 Training

The network is trained with gradient descent optimization algorithm with Nesterov momentum. The momentum is set to 0.99. The initial learning rate is 2e−4 and is halved after each preset boundary epoch, which is 0.2, 0.4, 0.6, 0.75, 0.8, 0.85, 0.9 and 0.95 of the total number of training epochs. L2 regularization is used to prevent overfitting with a weight decay of 1e−3.

3.3 Evaluation and Conclusion

The results are evaluated according to three metrics: Dice coefficient (Dice), 95th-percentile Hausdorff distance (HS) and Volumetric similarity (VS). Additionally, a sum of weighted metrics is computed as the overall score for MRBrainS ranking. Details of the evaluation metrics and the overall score are described in [18].

To compare the performance of three network variants, we run the leave-one-out cross validation as a 3-label segmentation problem (GM, WM and CSF) on the MRBrainS2018 training dataset. As shown in Table 2, MixNetv3 gives the best results. The cross validation results of MixNetv1 and MixNetv2 are quite close. But MixNetv2 has a lower validation loss (see Fig. 7). As discussed in Sect. 2.2, MixNetv1 contains the solution space of MixNetv3. However, the results of MixNetv1 is worse. We conjecture that the architecture of parallel modality streams can learn complementary features more easily.

By MixNetv2_multi, three classifiers are trained on the sagittal plane, the coronal plane and the transverse plane, respectively. Results are obtained by fusing predictions of three MixNetv2 classifiers with the corresponding weights 1:1:4. The weights are empirically chosen based on the fact that the transverse plane resolution is 4 times higher. Although the classifiers on the sagittal plane and the coronal plane performs much worse, the fused results still improves.

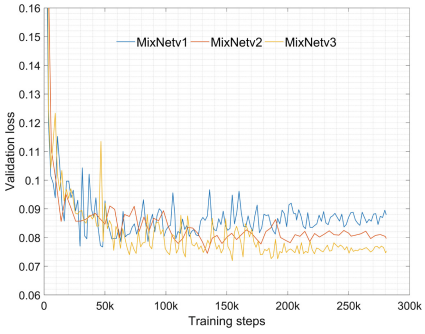


Fig. 7. Validation loss during training (subject 1 as the validation data).

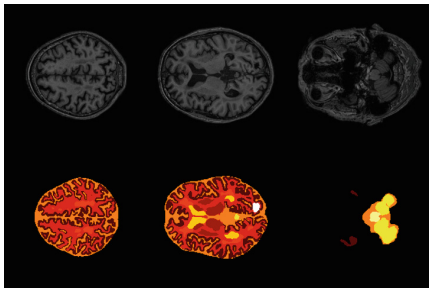


Fig. 8. Qualitative segmentation results of 8 brain structures.

MixNetv2_multi was also trained with the full training dataset as a 3-label and 8-label task. Figure 8 shows the qualitative results of 8-label predictions by MixNetv2_multi. Trained models were submitted to the challenge. Figures 9 and 10 show the evaluation performed by the challenge organizer on the test dataset. We notice a performance drop between the validation results and the evaluation results (about 0.02). That is reasonable, because the relatively small training dataset may not cover all the cases very well.

In the 8-label segmentation task, our network has difficulties in classifying WHM and basal ganglia. One possible reason is the imbalance of labels in the training data. We do not use any methods of balancing the labels during training, that is, labels with a small number in the training data are easy to ignore. The

Table 2. Cross validation results of MixNetv1, MixNetv2 and MixNetv3, performed on the MRBrainS2018 training dataset. The network is trained as a 3-label segmentation task (WM, GM and CSF).

	GM			WM			CSF		
	Dice	HD	VS	Dice	HD	VS	Dice	HD	VS
MixNetv1	.8524	.9583	.9728	.9000	1.9167	.9759	.8599	1.9167	.9508
MixNetv2	.8500	.9583	.9772	.8966	1.9167	.9626	.8609	1.9167	.9506
MixNetv2_multi	.8511	.9583	.9762	.9001	1.3553	.9689	.8624	1.9167	.9447
MixNetv3	.8557	0.9583	.9789	.9049	1.3552	.9743	.8609	1.9167	.9578

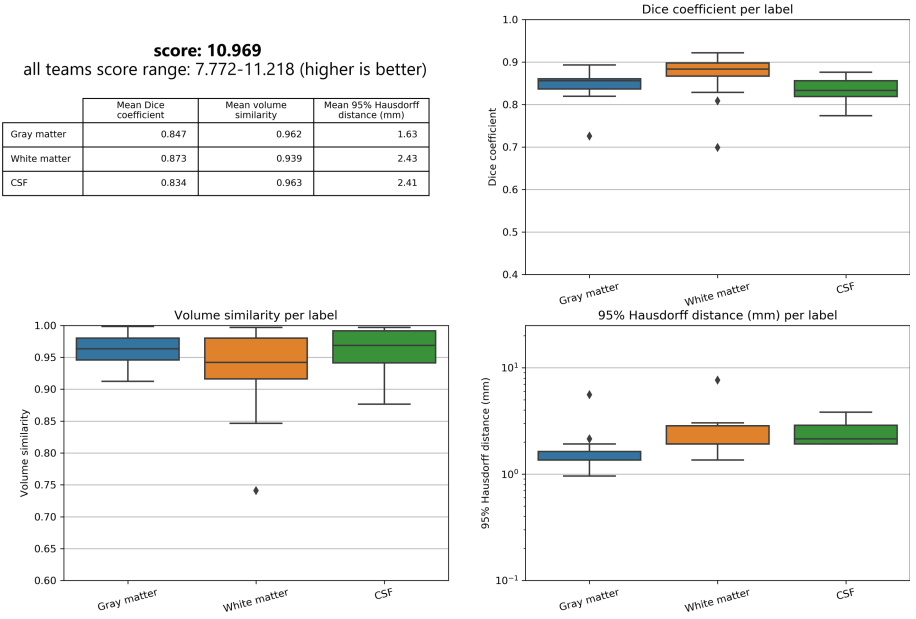
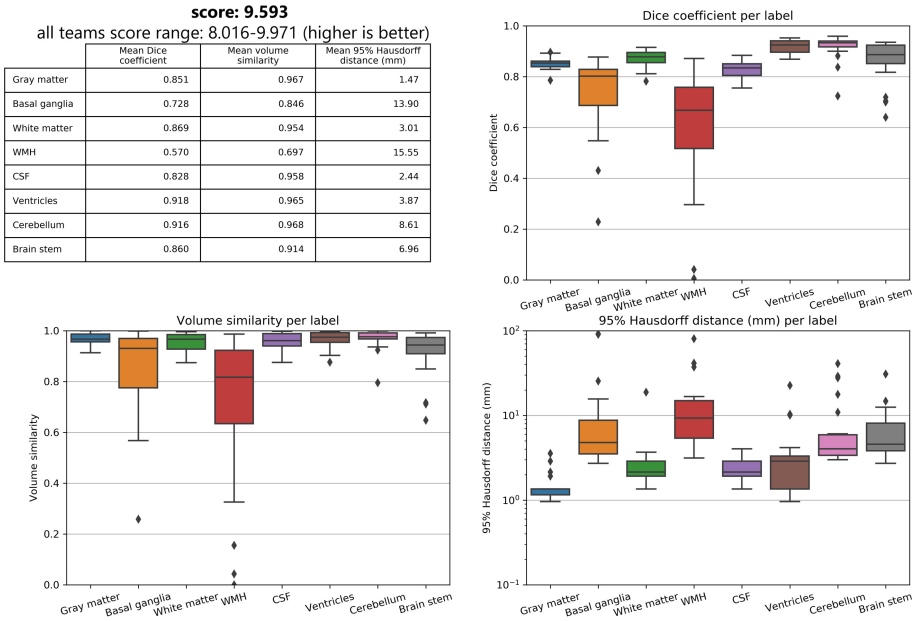


Fig. 9. Test results of MixNetv2_multi on MRBrainS2018 test dataset (3-label task).

8-label methods taking part in the MRBrainS2018 challenge differ mainly in the performance of segmenting WHM and basal ganglia. This problem deserves further study.



score: 9.593

all teams score range: 8.016-9.971 (higher is better)

	Mean Dice coefficient	Mean volume similarity	Mean 95% Hausdorff distance (mm)
Gray matter	0.851	0.967	1.47
Basal ganglia	0.728	0.846	13.90
White matter	0.869	0.954	3.01
WMH	0.570	0.697	15.55
CSF	0.828	0.958	2.44
Ventricles	0.918	0.965	3.87
Cerebellum	0.916	0.968	8.61
Brain stem	0.860	0.914	6.96

Fig. 10. Test results of MixNetv2_multi on MRBrainS2018 test dataset (8-label task).

4 Summary

In this work, we propose the MixNet, a deep residual CNN to tackle the brain structure segmentation problem. The network achieves state-of-the-art results with a relatively small training dataset. Three variants of MixNet is tested to see the effect of different modality mixing styles. Based on the experiment results, the network of parallel modality streams shows better performance, which implies that learning complementary features may be easier for this architecture.

As future work, a single classifier trained with images acquired along different orientations of the 3D MRI volume is worth testing. To do this, either a dataset of isotropic annotation resolutions is available or the resolution difference is tackled properly.

References

1. Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D.L., Erickson, B.J.: Deep learning for brain MRI segmentation: state of the art and future directions. J. Digit. Imaging **30**(4), 449–459 (2017)
2. Dale, A.M., Fischl, B., Sereno, M.I.: Cortical surface-based analysis: I. Segmentation and surface reconstruction. NeuroImage **9**(2), 179–194 (1999)
3. Fischl, B., Sereno, M.I., Dale, A.M.: Cortical surface-based analysis: II: inflation, flattening, and a surface-based coordinate system. NeuroImage **9**(2), 195–207 (1999)
4. Jenkinson, M., Beckmann, C.F., Behrens, T.E.J., Woolrich, M.W., Smith, S.M.: FSL. NeuroImage **62**(2), 782–790 (2012)
5. Ashburner, J., Friston, K.J.: Unified segmentation. NeuroImage **26**(3), 839–851 (2005)
6. De Brébisson, A., Montana, G.: Deep neural networks for anatomical brain segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 20–28 (2015)
7. Zhang, W., et al.: Deep convolutional neural networks for multimodality isointense infant brain image segmentation. Neuroimage **108**, 214–224 (2015)
8. Moeskops, P., Viergever, M.A., Mendrik, A., De Vries, L.S., Benders, M.J.N.L., Isgum, I.: Automatic segmentation of MR brain images with a convolutional neural network. IEEE Trans. Med. Imaging **35**, 1252–1261 (2016)
9. Nie, D, Li, W, Gao, Y, Sken, D.: Fully convolutional networks for multi-modality isointense infant brain image segmentation. In: 13th IEEE International Symposium on Biomedical Imaging (ISBI), pp. 1342–1345 (2016)
10. Shin, H.-C., et al.: Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging **35**(5), 1285–1298 (2016)
11. He, K., Zhang, X., Ren, S., Sun J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE (2016)
12. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
13. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
14. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
15. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
16. Tustison, N.J., et al.: N4ITK: improved N3 bias correction. IEEE Trans. Med. Imaging **29**(6), 1310–1320 (2010)
17. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, p. 958. IEEE (2003)
18. MRBrainS2018 Homepage. http://mrbrains18.isi.uu.nl/. Accessed 11 Oct 2018