Training CNNs for Multimodal Glioma Segmentation with Missing MR Modalities

Karin van Garderen[1] K.VANGARDEREN@ERASMUSMC.NL
Marion Smits [1] M.SMITS@ERASMUSMC.NL
Stefan Klein [1,2] S.KLEIN@ERASMUSMC.NL
[1] *Erasmus MC, Dept. of Radiology and Nuclear Medicine, Rotterdam, the Netherlands*
[2] *Erasmus MC, Dept. of Medical Informatics, Rotterdam, the Netherlands*

Editors: Under Review for MIDL 2019

Abstract

Missing data is a common problem in machine learning, and in retrospective imaging research it is often encountered in the form of missing imaging modalities. We propose to take into account missing modalities in the design and training of neural networks, to ensure that they are capable of providing the best possible prediction even when one of the modalities is not available. This would enable algorithms to be applied to subjects with fewer available modalities, without leaving out the same information in other subjects or applying data imputation. This concept is evaluated in the context of glioma segmentation, which is a problem that has received much attention in part due to the BraTS multi-modal segmentation challenge. The UNet architecture has been shown to be effective in this problem and therefore it serves as the reference method in this paper. To make the network robust to missing data we leveraged the dropout principle during training and applied this to the UNet architecture, but also to variations on the UNet architecture inspired by multimodal learning. These networks drastically improved the performance with missing modalities, while only performing slightly worse on the full dataset.

Keywords: convolutional neural network, glioma segmentation, multimodal

1. Introduction

Tumor segmentation is a key task in brain imaging research, as it is a prerequisite for obtaining quantitative measures of the tumor. Since manual segmentation by radiologists is time-consuming and prone to inter-observer variation, there is a clear need for effective automatic segmentation methods. Research into these methods for glioma has been accelerated by the recurring BraTS multi-modal segmentation challenge on low-grade glioma (LGG) and glioblastoma (GBM) (Menze et al., 2015). Each iteration of the challenge has brought better performing segmentation methods, with the best performing methods in recent editions all based on convolutional neural networks (CNNs).

While the BraTs challenge focuses on improving performance, there are practical problems to overcome before automatic segmentation can be applied in practice. One of these challenges is dealing with missing data. The BraTS benchmark contains four MR modalities: a T1-weighted image (T1W), a T1-weighted image with contrast agent (T1W+C), a T2-weighted image (T2W) and a FLAIR image (FLAIR), which are co-registered so that corresponding voxels in the image are aligned. Each of these images provides a different

piece of information to the radiologist and a CNN can learn to segment a tumor from the combination of modalities. Although these images are complementary, a radiologist is still able to perform a decent segmentation if one of these modalities is missing while this is not guaranteed for a CNN. Especially in retrospective studies it is not unlikely that the imaging protocol does not contain all four modalities and even if it does, poor image quality or registration errors can cause similar problems.

1.1. Dealing with missing data

A common way to deal with missing data is to use imputation, to provide the network with some substitute value that is common (e.g. the mean or median) or even a value that is generated from the remaining data. Artificial neural networks can also be used to generate missing data (Jerez et al., 2010), and in the specific case of medical imaging they could be used to generate entire images. However, we propose to avoid imputation completely by designing and training a CNN so that it can adapt to missing data naturally.

In this paper we explore segmentation networks that are inherently robust to missing data by leveraging the dropout principle (Srivastava et al., 2014), either through a simple adaptation to the training procedure or by adapting the CNN architecture completely. Dropout layers are a way of regularizing the network by randomly removing features during training, which is supposed to make features more robust. In CNNs dropout is commonly implemented on feature channels, which means that entire kernels are removed instead of single features. In this study dropout layers are applied specifically to the layers where image modalities are merged, so that the networks become more robust to missing data.

1.2. Combining modalities in CNNs

In most segmentation methods the modalities are concatenated as input channels to the CNN, as if they were color channels in an RGB image (Pereira et al., 2016)(Işın et al., 2016). This way of incorporating different images enables the network to learn low-level features from the combination of MR sequences, which makes sense if we assume that the voxel intensity in the different modalities correspond to different features of the same tissue. However, this also makes the network vulnerable to missing data, since all features depend on all input modalities to some extent. We explore adapted network architectures where the information from different sequences is merged at a much later stage in the network, so that the majority of features rely only on a single modality.

Although these architectures lack the ability to learn low-level features from multiple MR sequences, they do offer potential benefits in terms of robustness and trainability. They are potentially more robust to missing data and errors which occur in one of the input modalities, such as registration errors, poor image quality or artifacts, because these errors can not affect the low-level features from other modalities. Also, a network with separate pathways for each modality offers the possibility to train these pathways individually and this means that datasets with a one or more missing modalities can be used for training as well. Furthermore, if these pathways are trained separately, it is easier to increase the size of the network while keeping the memory usage limited.

2. Methodology

2.1. Network architecture

The 3D UNet architecture (Çiçek et al., 2016) is a well-established segmentation network and still was one of the best performing architectures at the most recent 2018 BraTS challenge (Isensee et al., 2018). Therefore the UNet forms the baseline for our research as a reference network and a building-block for different variations of multi-modal architectures. The number of trainable parameters in the model depends on the number of feature maps in each convolution, which we chose to parameterize by a single variable c. The first convolution has c kernels, and as the size of the feature maps decreases the number of kernels is increased. Figure 1 shows the UNet architecture with the number of feature maps per convolution layer expressed as a multiple of c.

In the reference UNet architecture each 3D convolution block contains a batch normalization, a 3D convolution layer with kernels of size 3^3 and ReLu activation. The last fully connected layer was implemented as a 3D convolution with kernels of size 1^3. The downsampling step is a max-pooling layer of stride 2 and size 2^3 and the upsampling was implemented by tri-linear interpolation. For this UNet architecture each target voxel has a receptive field of 88^3 voxels.

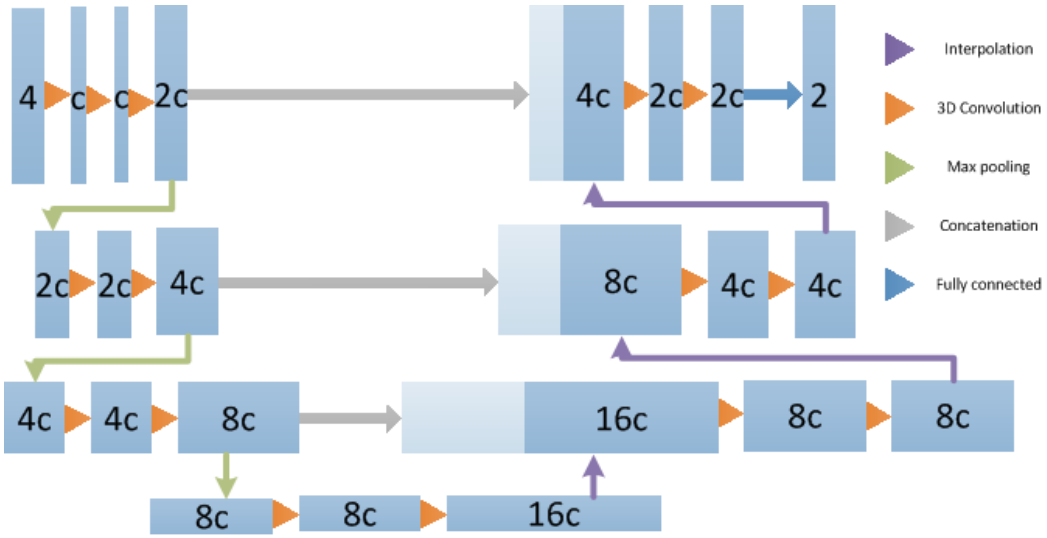

Figure 1: Illustration of the UNet architecture. The number of feature maps, as a function of the parameter c, is indicated for each step.

UNet with Dropout

A simple way to make the network robust to missing data, is to train it with missing data. Therefore the UNet architecture was adapted with a single dropout layer at the input, replacing the Batch Normalization layer, which removes entire input channels (MR

sequences) with probability $p = 0.2$. This dropout probability is lower than in the multi-pathway architectures (see below) because it has fewer feature channels.

Ensemble network

The Ensemble and Late Fusion architectures explore a completely different way of combining the MR modalities. In these architectures, one network is trained for each MR modality and the predictions (or features) of each of them is combined in a final prediction layer. The ensemble network architecture applies a technique similar to the concept of ensemble learning, where multiple algorithms or instances of the same algorithm are combined after training them separately (Maji et al., 2016). In this implementation, the separate UNet pathways produce probability estimates per class, which are concatenated to eight channels in the fusion layer. The final prediction layer is a 1^3 convolutional layer with dropout ($p = 0.5$), so the final layer is in fact a trainable combination of all predictions.

Late Fusion network

The idea of a fusion network originates from multimodal learning (Ngiam et al., 2011), where different modalities are more inherently different, such as images and text. The network has one pathway for each of the four modalities and the feature maps of the final convolutional layer are concatenated to an output of $8c$ channels. The final prediction is performed again by a 1^3 convolution layer with dropout ($p = 0.5$). The late fusion network can be trained end-to-end or separately. When trained separately, a UNet is trained for each individual MR modality and the separate prediction layers are replaced by the fusion layer followed by a single prediction layer.

The last layer of the Ensemble and Late Fusion network needs to be trained, but the original UNet weights can be frozen while doing this which drastically decreases the required memory with respect to training the full network end-to-end. The networks are visualized in Figure 2.

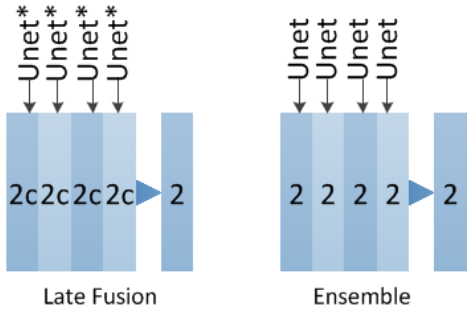

Figure 2: Illustration of the late fusion network (left) and ensemble network (right). The asterisk (*) indicates that the last feature layer is used instead of binary label predictions.

2.2. Data

The networks were trained and evaluated on the training set of the BraTS challenge 2018 (Bakas et al., 2017), which is a benchmark dataset of pre-operative scans of 278 patients with low-grade glioma (LGG, 75) or glioblastoma (GBM, 203). The images in this benchmark are skull-stripped, co-registered and resampled to a size of 240 by 240 by 155 voxels. The segmentations contain separate labels for the tumor edema, core and enhancing core, but we restricted the analysis to a binary whole-tumor segmentation.

2.3. Preprocessing

The non-background voxels of each separate image were normalized to zero mean and unit standard deviation. To limit memory usage, random patches of 108^3 voxels were extracted, which correspond to 20^3 target voxels. With a probability of 50% a patch was selected from a tumor area, meaning that the center voxel was part of the tumor, and with 50% probability the center voxel was located outside of the tumor but inside the brain.

2.4. Training and evaluation

The networks were optimized with the Adam optimizer (Kingma and Ba, 2014) and the cross-entropy loss function. For every epoch, one patch from each patient was extracted randomly and fed to the network. The dataset was divided into five cross-validation folds, so that 20% of the subjects were always selected for testing and never used during training. The folds are random, but the same for each experiment. Due to time constraints, only two of these folds were evaluated.

Evaluation took place on the whole image, although it was classified by the network in patches to limit memory usage. These patches were selected to cover the whole image. The Dice coefficient is reported as performance metric.

2.5. Evaluating robustness

To assess whether the UNet with Dropout, the Ensemble network and Late Fusion network are indeed more robust to missing data, we evaluated the same models in a situation where one of the sequences is removed. The result is expected to be largely dependent on the sequence that is affected, so the four sequences were evaluated separately. The missing sequence was removed from the network by setting the input to zero and scaling the other inputs by $n/(n-1)$, where n is the number of original inputs, which is equivalent to the procedure applied in dropout layers. For the single UNet architecture this procedure was applied at the input, but for the Ensemble and Late Fusion networks the dropout procedure was applied at the fusion layer.

2.6. Network parameters and implementation

For a fair comparison of the different architectures it is important to consider the number of trainable parameters. For the Ensemble and Late Fusion architectures, if c were the same, the network would have approximately four times the number of trainable weights. To create a network of the same size as a single reference network the UNets that form the pathways of a multi-pathway network should have half the number of channels per layer,

because the number of weights scales quadratically with c. Taking this into account, we designed and trained the different networks in the following way:

- **UNet**: A single UNet with four input channels and $c = 32$. It was trained end-to-end for 300 epochs. The number of trainable weights is $16,323,690$.
- **UNet dropout**: A single UNet with four input channels and $c = 32$, where dropout was applied to input channels with $p = 0.2$. It was trained end-to-end for 300 epochs. The number of trainable weights is the same as UNet.
- **Ensemble**: An ensemble network of four UNets and $c = 16$. The networks were trained separately with a single MR sequence (200 epochs). The final prediction layer was added and trained for 100 epochs while the rest of the weights were frozen. The number of trainable weights is $16,329,890$.
- **Late Fusion**: A late fusion network with four pathways in the shape of a UNet ($c = 16$). The networks were trained separately with a single MR sequence (200 epochs). The final prediction layer was added and trained for 100 epochs. The number of trainable weights is $16,330,130$.

All networks were implemented in PyTorch (Paszke et al., 2017) and trained on two NVIDIA GeForce GTX 1080 Ti graphics cards, each with 11 Gb memory.

3. Results

The four networks were evaluated with the full datasets and after removing each of the four modalities, on two cross-validation folds of 57 subjects each. The results are summarized in Table 1 and the distribution of performance across subjects is shown using boxplots in Figure 3.

Figure 3 shows that, on the full dataset (without missing data), the simple UNet without dropout performs best, but the other networks are not significantly worse. For missing data scenarios, the regular UNet suffers while the other networks are able to maintain a decent performance. It seems that FLAIR is the most informative MR sequence in this problem, and this is not surprising because that image was used as gold standard for the original whole-tumor segmentation. The Ensemble network is most stable in its performance and suffers the least from missing data, but it is also the least effective when using the full dataset.

Significance between results was tested using the Wilcoxon signed-rank test and reported in Table 1 for the comparison of different methods to the UNet baseline. The difference between the UNet with dropout and the Ensemble networks is also significant in the case of missing FLAIR image ($p < 0.001$ for both folds).

4. Discussion and conclusion

We showed that it is possible to design and train a CNN to be robust to missing MR modalities, in the context of the BraTs multi-modal segmentation challenge with four MR

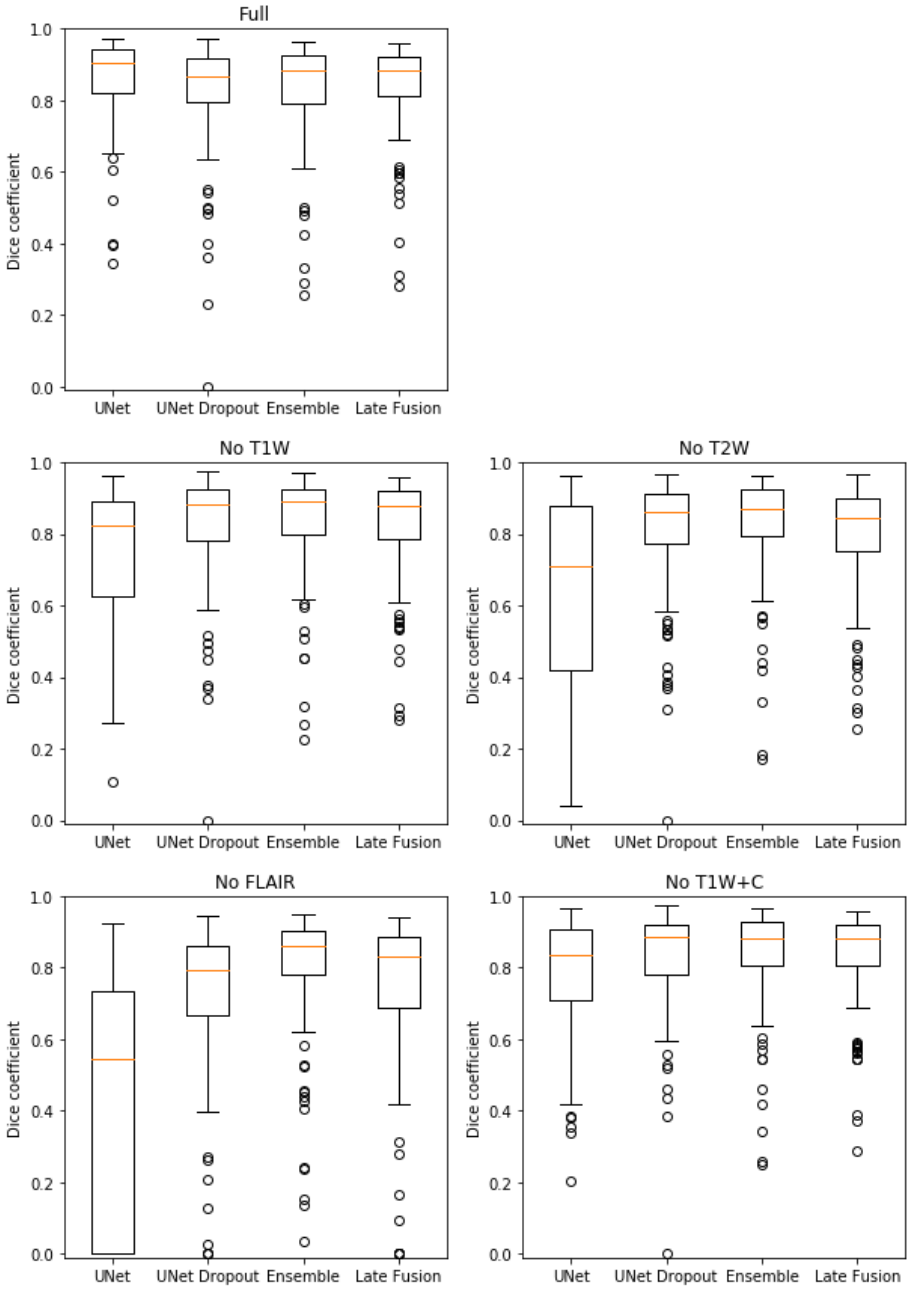

Figure 3: Boxplot of performance on both cross-validation folds for each network, with full data and single MR sequences removed.

Table 1: Numeric results for in terms of mean Dice coefficient and standard deviation over subjects. Results with asterisk are significantly different from the corresponding UNet performance ($p < 0.001$).

	Model	Full	No T1W	No T2W	No FLAIR	No T1W+C
Fold 1	UNet	**0.86 (0.13)**	0.78 (0.20)	0.78 (0.19)	0.56 (0.31)	0.80 (0.16)
	UNet Dropout	0.85 (0.13)	0.82 (0.17)*	0.82 (0.16)	0.73 (0.22)*	**0.85 (0.12)***
	Ensemble	0.85 (0.15)	0.84 (0.15)	**0.84 (0.15)**	**0.81 (0.18)***	0.85 (0.14)
	Late Fusion	0.85 (0.12)	**0.84 (0.14)***	0.80 (0.18)	0.80 (0.16)*	0.85 (0.12)*
Fold 2	UNet	**0.85 (0.11)**	0.71 (0.18)	0.50 (0.27)	0.29 (0.32)	0.78 (0.16)
	UNet Dropout	0.79 (0.18)*	0.83 (0.15)*	0.79 (0.18)*	0.72 (0.21)*	0.81 (0.17)
	Ensemble	0.83 (0.14)	**0.83 (0.14)***	**0.81 (0.15)***	**0.78 (0.18)***	**0.83 (0.14)**
	Late Fusion	0.82 (0.14)	0.82 (0.15)*	0.79 (0.13)*	0.69 (0.25)*	0.82 (0.14)

sequences. Applying dropout on the input channels is a simple way to achieve robustness with only a minimal impact on performance for complete datasets. More advanced multimodal architectures, with a separate pathway for each modality, might give an even better balance between performance and robustness with the additional benefit of training the pathways separately. Especially in retrospective studies, where the availability of different modalities is not guaranteed, these architectures simplify the training procedure by considering the modalities separately.

One important parameter to take into account is the dropout percentage. In this study it is set to $p = 0.5$ for the multi-pathway architectures, but for the UNet input channels it was set to $p = 0.2$ due to the smaller number of channels in the dropout layer. Further research is needed to assess the effect of the dropout percentage on performance and robustness.

From this study it seems that the Ensemble network shows the best balance between overall performance and robustness. However, this could be due to the balance between dropout probability and the number of feature maps in the fusion layer, which is eight for this network and $8c = 128$ for the Late Fusion network. For the Ensemble network it is likely that the information from an entire MR modality is sometimes removed, while this is highly unlikely for the Late Fusion network. A logical next step would be to design a specialized dropout layer that removes specific subsets (i.e. entire pathways) of the input rather than random elements.

In this paper we consider a specific use-case, but the principle of leveraging dropout for missing data extends to other applications of deep learning in medical imaging. Most of all this paper is a proof of concept, demonstrating that it is worthwile to consider missing data when training an algorithm and that neural networks offer the possibility to be inherently robust to missing data. This prevents researchers who need the algorithms in practice from having to apply strict selection criteria based on the available data or use data imputation. Further research on different architectures and training procedures can possibly improve the performance and robustness even more.

Acknowledgements

This work was supported by the Dutch Cancer Society (project number 11026, GLASS-NL).

References

S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J.S. Kirby, J.B. Freymann, K. Farahani, and C. Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117, 2017.

Ö. Çiçek, A. Abdulkadir, S.S. Lienkamp, and O. Brox, T.and Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing.

A. Işın, C. Direkoğlu, and M. Şah. Review of mri-based brain tumor image segmentation using deep learning methods. *Procedia Computer Science*, 102:317 – 324, 2016. 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016.

F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein. No New-Net. *ArXiv e-prints*, September 2018.

J.M. Jerez, I. Molina, P.J. García-Laencina, E. Alba, N. Ribelles, M. Martín, and L. Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine*, 50(2):105 – 115, 2010. ISSN 0933-3657.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

D. Maji, A. Santara, P. Mitra, and D. Sheet. Ensemble of deep convolutional neural networks for learning to detect retinal vessels in fundus images. *CoRR*, abs/1603.04833, 2016. URL http://arxiv.org/abs/1603.04833.

B. H. Menze, A. Jakab, S. Bauer, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A.Y. Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

S. Pereira, A. Pinto, V. Alves, and C.A. Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.