Brain Tumour Segmentation with Random Forest and U-Net

Alexander Kugele
Heidelberg University
Matrikel-Nr: 3150330
Physics, M.Sc.
a.kugele@stud.uni-heidelberg.de

Shuhan Xiao
Heidelberg University
Matrikel-Nr.: 3160697
Physics, M.Sc.
shuhan.xiao@stud.uni-heidelberg.de

Abstract

This report compares two methods to segment 2D brain tumour images: A random forest and a U-Net. It is found that without manual tuning, both methods are comparable in their segmentation accuracy, achieving Dice scores of 65.9% and 69.3% respectively. However, training and inference time of the U-Net are significantly smaller and training larger datasets is possible more easily.

1 Introduction

An important task of image processing and image analysis is semantic segmentation, which assigns a class label to each pixel of an image by combining information about the intensity value of the pixel itself as well as its adjacent pixels. One of the most challenging application of image segmentation is the segmentation of biomedical images. It faces the problem that often the appearance of the objects of interest can be highly variable. Especially in medical imaging there might also be variations in image quality, for example caused by motion artefacts and intensity inhomogeneity.

This project focusses on the segmentation of gliomas, which are the most common type of tumours, accounting for 81% of all malignant tumours found in both children and adults (Ostrom et al., 2014). Gliomas are categorised into low-grade gliomas (LGG) and high-grade gliomas (HGG) according to their rate of growth and prognosis for the patient.
The tumours are usually diagnosed using magnetic resonance imaging (MRI). The advantage of MRI is that it is a non-invasive imaging technique. Additionally, different image contrasts by using different acquisition techniques, i.e. types of scans. Commonly, T_1-weighted (T1), T_1-weighted contrast enhanced (i.e. using the contrast agent gadolinium) (T1c), T_2-weighted (T2) and Fluid-Attenuated Inversion Recovery (FLAIR) images are acquired in multimodal MRI protocols and used for diagnosis, as they provide information about different properties about the tissue.
A challenge of segmenting gliomas is that the tumours are highly heterogeneous, often have no clear defined edge and have variable sizes, shapes and locations. The boundaries of LGG tumours are particularly difficult to detect. This is why the usually trained professionals are required to manually label the images. The labelling is however time consuming and not reproducible.
Another reason why tumour segmentation is essential is its many applications:

- diagnosis
- categorisation of tumour and state of disease
- prognosis (e.g. to predict survival)
- localisation of tumour for treatment (radiotherapy, surgery)
- monitoring the development and growth

Tumour segmentation is required to have a high accuracy. It is also important that it is quantitative, automatable and reasonably fast.
Numerous of techniques have been developed for the task of segmentation so far. The goal of this student project is to adapt two of those techniques to our problem of tumour segmentation and to the hardware available to us: the random forest and the U-Net.
The random forest (Ho, 1995) is an ensemble methods, combining several decision trees. It developed to deal with mulitclass problems and was shown to be robust outliers and less likely to overfit.
The U-Net was originally applied the to cell tracking (Ronneberger et al., 2015). It is a type of fully convolutional network and specifically developed for segmenting biomedical image.
These techniques are further described in section 4. We have applied our methods to the Multimodal Brain Tumor Image Segmentation Benchmark (BRATS 2015), which is further described in section 3. Our results and comparison of the two techniques are presented in section 6.

2 Related Work

Both random forest and U-Nets have been previously applied to segment tumours in the BRATS challenge.

Pinto et al. (2015) applied their extremely randomized forest to the BRATS 2013 dataset using both appearance- and context-based features and achieved a Dice score of 83% and a sensitivity score of 83% for the whole tumour region.
Random forests were also used by Maier et al. (2015), who obtained a Dice and sensitivity score of 84% and 85% respectively for the whole tumour and only LGG cases. Their features include the intensity, gaussian, local histogram, hemispheric difference and center distance.
A U-net combined with data augmentation has been previously used in the BRATS 2015 challenge by Dong et al. (2017) successfully. They achieved a Dice score of 81% for the complete LGG tumours.
With an an ensemble of a DeepMedic architecture consisting of fully 3D multi-scale convolutional neural networks (CNN), three fully convolutional networks (FCN), and two 3D U-Nets, Kamnitsas et al. (2017) obtained the best overall result of the most resent BRATS challenge in 2017 (Bakas et al., 2017) with their so called Ensembles of Multiple Models and Architectures. Their method lead to a Dice score of 90.1% and a sensitivity score of 89.5% for the whole tumour.

3 The Dataset

3.1 General Information

Image segmentation is primarily done in two ways: Either designing an algorithm from first principles or using available data to train an algorithm. As for this challenge, designing an algorithm without data is very difficult, the method of choice is training an algorithm from data. For the Multimodal Brain Tumor Image Segmentation Benchmark (Menze et al., 2015) in 2015, 3D MR tumour scans where collected from the BRATS 2012 and BRATS 2013 challenge and from the NIH Cancer Imaging Archive (TCIA).
In total, data for 55 low-grade glioma patients and 220 high-grade glioma patients is provided. The data is a 16-bit 3D scan of shape (depth=155, height=240, width=240), where all datasets have been aligned to the same anatomical template and interpolated to 1 mm^3 voxel resolution. For each patient, four scan types are available: T1, T1c, T2 and FLAIR. In the case of the BRATS data, the labels are from expert annotations of one to four raters. The TCIA data labels were obtained by fusing the results of multiple segmentation algorithms from the BRATS 2012 and BRATS 2013 challenge and reviewed by expert raters.
Four classes are predefined:

0) background
1) necrosis
2) edema
3) non-enhancing tumour
4) enhancing tumour

Each 3D image for each scan type is saved in a separate `.mha` file. The average size per file is about 2.2 MB, leading to approximately 2.2 MB $\cdot 275 \cdot 4 = 2420$ MB of compressed data. Uncompressed, the data size increases to over 100 GB. In fig. 1, examples for LGG and HGG and the different scan types are shown.

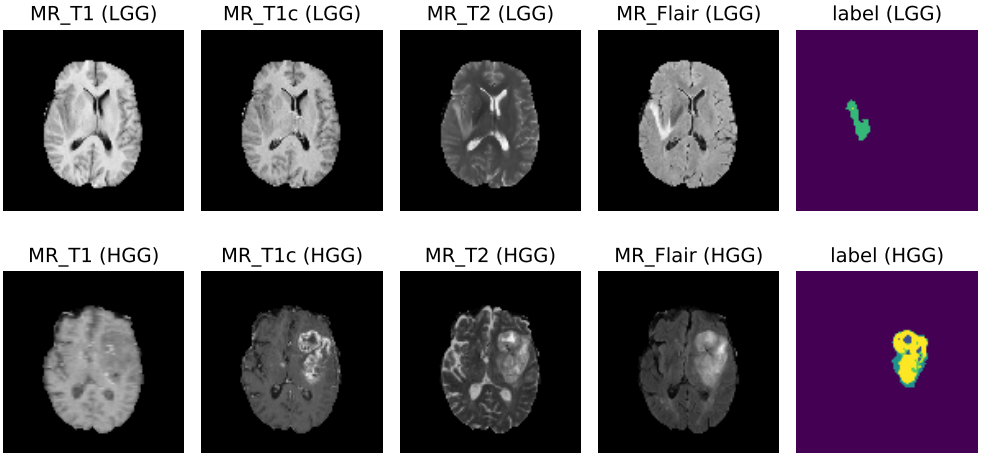


Figure 1: Dataset examples. *Top:* Low-grade glioma tumour. The structure is primarily visible in the Flair scan. *Bottom:* High-grade glioma tumour. The structure is richer and the different scan types help in differentiating the different types of tumour.

3.2 Utilizing the Dataset

The given images are sliced 3D scans. As we want to do segmentation in two dimensions, we just take each slice as a separate input image. In total, this are $155 \cdot 275 \cdot 4 = 170500$ input images. This full dataset proved too be to large to use it for training. After trying out multiple subsets, it was decided to use only the LGG part of the dataset and to sort out images where the tumour to background ratio is less than 0.1%. We only used FLAIR images, which provide the most meaningful information about the tumour structure as illustrated in fig. 1. Registration, which would have been inaccurate due to the varying resolutions of the scans, was therefore not necessary.

This dataset only takes 110 MB compressed on disk. We split this dataset of 3315 images further up into a training set of 2652 images (80%) and a test set of 663 images. This dataset is used for both the random forest and the U-Net. The classes are reduced from five to two: background and tumour. Loading 10 images in the memory of the GPU for training takes about 3 GB. For the random forest, using 1000 images at once takes about 30 GB of RAM.

4 Methods

4.1 Random Forest

The random forest is a set of decision trees, where each tree is fitted to a random subset of the data and predictions are made by taking averages over the predictions of individual trees. More precisely, given N data points, N points are sampled at random with replacement from the data. These are then used to train a decision tree. This kind of sampling is called bagging. The samples that are not used for the training of a decision trees are so-called out-of-bag samples. These can be used to test the generalization accuracy of the training.

Training of each decision tree in scikit-learn is done with the Classification and Regression Tree (CART) algorithm. Before explaining the training algorithm, the Gini impurity is explained. This

measure is used to determine the next split in a tree. Generally, it can be calculated as

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2 \tag{1}$$

where i is the node index, k is the class label and $p_{i,k}$ is the ratio of class k to all classes in the node. The CART algorithm uses a loss based on this measure

$$J(k, t_k) = \frac{m_L}{m} G_L + \frac{m_R}{m} G_R \tag{2}$$

such that it finds for each current leaf a class and a threshold such that the impurity is minimized for the left and the right leaf to be added. This is done until the impurity cannot be further reduced or the maximum depth is reached. The default parameter in scikit-learn is `max_depth=None`, meaning it goes as deep as it can. This can lead to large trees. An example of one tree after training is shown in section 9.2, where each small orange or blue dot is a single node.

4.1.1 Features

A random forest classifies each pixel separately. Therefore, using only the pixel intensities would only give a good classification if the tumour can be identified on a single-pixel level. To include information about the local structure, multiple features are defined, such that each input pixel is an N-dimensional vector.
The features we chose for this task are:

1. Gaussian filter
   - Convolution with a Gaussian kernel
   - Smoothes the image
2. Laplacian of Gaussian (LoG) filter
   - Convolution with the Laplacian of a Gaussian kernel
   - highlights edges (edges are 0)
3. Gaussian gradient magnitude filter
   - Convolution with the gradient of a Gaussian kernel
   - highlights edges (edges are extrema)
4. Eigenvalues of the Hessian matrix
   - determines the surface concavity
5. Eigenvalues of the structure tensor
   - Summarizes gradient direction
6. Equalized histogram
   - Enhances contrast

Except for the equalized histogram, they all share the tuneable kernel width σ, which is set to $\{0.3, 0.7, 1., 3.5\}$. In total, $N = 29$ filters are used. In fig. 2, the features are shown for one example image of the dataset for $\sigma = 0.3$. The tumour region in the bottom left can be clearly identified for the Gaussian (bright), Laplacian of Gaussian (dark) and the equalized histogram (bright). However, for other examples the tumour is also pronounced in the other features.

In principle one could also utilize the different scan types as features. However, the dataset is already very big and therefore it was decided to not take the other scan types as features. This also allows for a fair comparison between the random forest and the U-Net, as they are then trained with the same dataset.

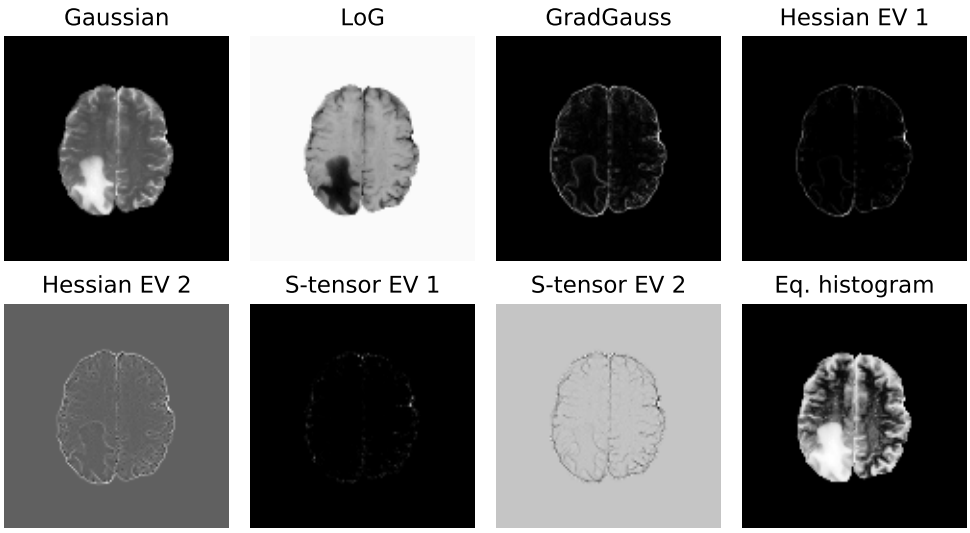


Figure 2: Features for the random forest. The Gaussian, Laplacian of Gaussian and the equalized histogram enhance the tumour region. In this example, the eigenvalues of the structure tensor and the Hessian matrix are not significant for the classification. However, they improved the classification on a small test set and therefore are also kept.

4.2 U-Net

We implemented a U-Net for our problem, which is based on a fully convolutional network by Long et al. (2014). Ronneberger et al. (2015) showed that the U-net requires only a few training images when combined with data augmentation. We based our implementation on the versions of Deng (2017) and Huang (2017), with a few additional modifications.

The distinct u-shaped architecture of the original U-Net with its skip-connections is shown in fig. 3. It has two symmetric paths: a contracting (down-sampling) and an expanding (up-sampling) path, each originally with five blocks.
We used a U-Net with the depth of only four blocks for each path instead to increase training speed and as the size of our input images was only 240×240. Our lowest resolution of the feature maps with 30×30 pixels is still smaller than the resolution of the original U-Net with 32×32 pixels. In total, our whole U-Net has only 18 convolutional layers instead of 23.

The contracting path allows to capture context of each pixel and to extract information from different feature levels. In each block two 3×3 convolutions with stride 1 are performed, each followed by rectified linear unit activation function (ReLU). In contrast to the original U-Net, the images are zero-padded in order to avoid cropping. We also added batch normalisation after each ReLU to improve the gradient flow, allowing higher learning rates and thus increase training speed. Max pooling with stride 2×2 is then applied for downsampling. This also doubles the number of feature channels up to 512 in after the fourth block, while reducing the spacial information. We used dropout in the forth block to decrease overfitting, as no data augmentation was used.
The expanding path is symmetric to the contracting path. In this path, the resolution is increased so that context information is propagated to higher layers. In each block, a concatenation with the corresponding feature map from the contracting path is performed, followed by upsampling with a 2×2 up-convolution, two zero-padded 3×3 convolutions with stride 1, ReLU activation and batch normalisation. Due to previous padding in the contracting path, no cropping is required here. The up-convolution halves the number of feature channels while increasing the size of the feature maps. The skip-connections between the two paths allows the combination of spatial information from the contracting path and information about the context from the expanding path.
In the final layer each feature map with 64 channels is mapped to the number of classes, which is 1

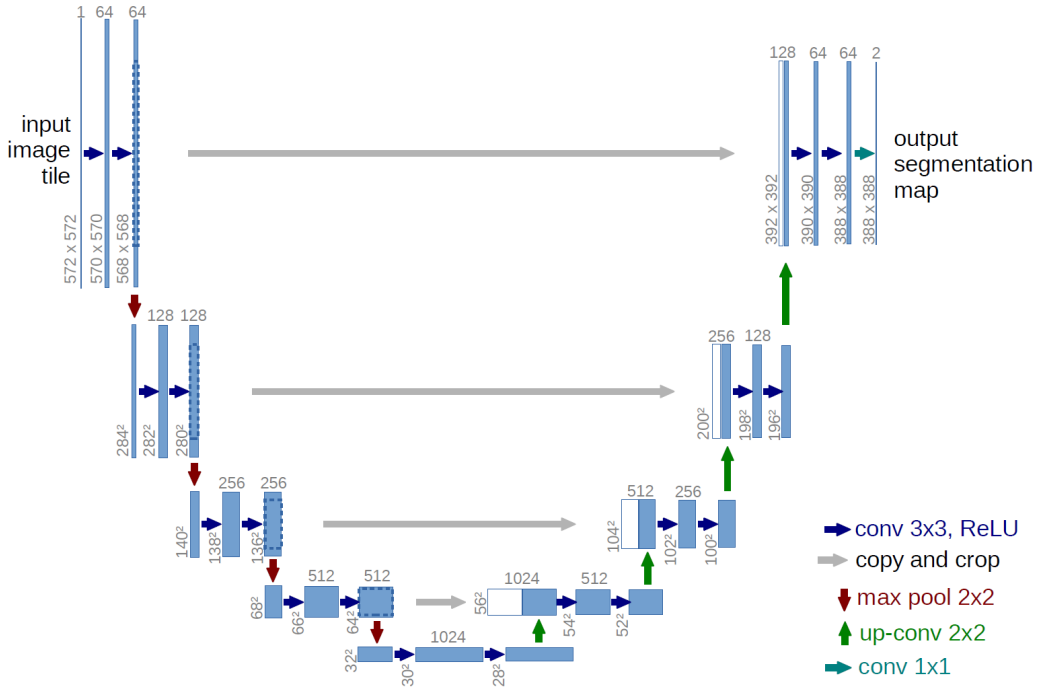


Figure 3: U-shaped architecture of the original U-net. (Ronneberger et al., 2015)

for the case of binary segmentation, by a 1×1 convolution.
After applying an argmax function, or in our binary case a sign function, the output of the network is a pixel-wise (binary) segmention of the input image.

5 Experiment

5.1 Training

5.1.1 Random Forest: Batch-mode training

Using the features defined in section 4.1.1, the complete training data still takes about 70 GB of memory. Therefore, it is decided to train the random forest sequentially on large chunks of the dataset as suggested in Ristin et al. (2016). The dataset is divided in three batches of about 26 GB each, assuming that in each large batch, the distribution of features is approximately equal. Then, we train a random forest on 100 estimators for each batch. The resulting random forest therefore has 300 estimators. It takes approximately 15 GB on disk, saved in the pickle format. In scikit-learn, the `warm_start=True` parameter of the Random Forest can be used to do this kind of batch-mode training.

Training the random forest took about 13 hours. Storing it on disk takes about 15 GB in pickled format. Both training time and classifier disk size can be reduced by setting adjusting the hyperparameters `max_depth`, `min_impurity_leaf`, `max_leaf_nodes`, `min_samples_split`, `min_samples_leaf`. However, finding a good setup is also time-consuming and we mainly want to compare the segmentation accuracy. Therefore, the standard values are kept. Therefore, loading the classifier takes about 310 s and a single prediction takes about 2-20 s.

5.1.2 Training of the U-Net

The U-Net was trained for 30 epochs with a batch size of 10 images per batch. The weights were intitialised by Xavier initialisation. Instead of the cross-entropy loss used by Ronneberger et al. (2015) we used a loss based on the Dice score (section 5.2), which was also our evaluation metric.

We used the Adam Optimizer in pytorch for faster training. Training took 1.5 hours in total with an inference time of 0.5 s per image and a loading time of 20 s. It took a disk space of 0.3 GB and a memory space of 0.7 GB during inference.

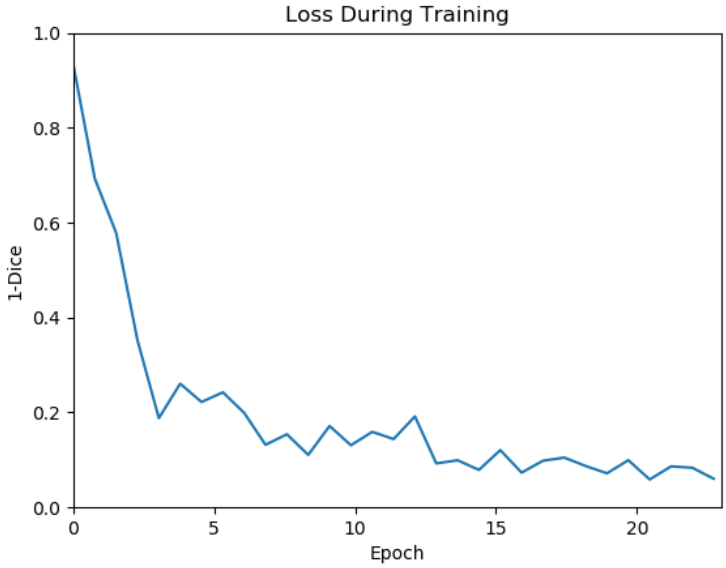


Figure 4: Loss of the U-Net during training. After 30 epochs a loss of approximately 0.1, corresponding to a Dice score of 0.9 for the training set, was achieved.

5.2 Evaluation Metrics

For a quantitative comparison we calculated three metric as described by Menze et al. (2015). We used the Dice score $\in [0, 1]$ for our evaluation given by

$$Dice = \frac{2|P_1 \cap T_1|}{|P_1| + |T_1|} \tag{3}$$

$$= \frac{2\,TP}{FP + 2\,TP + FN}, \tag{4}$$

where P_1 and T_1 represent the pixels corresponding to the tumour of the prediction and ground truth respectively. *TP*, *FP* and *FN* denote the true positive, false positive and false negative pixels respectively. The Dice score is a measure for the overlap between the tumour regions of prediction and ground truth. The loss of the U-Net was set to

$$Loss = 1 - Dice. \tag{5}$$

We also calculated the sensitivity score (true positive (*TP*) rate or recall) and the specificity score (true negative (*TN*) rate)

$$Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

and

$$Specificity = \frac{TN}{TN + FP}. \tag{7}$$

We used these scores because they are able to deal with the class imbalance. In the case of tumour segmentation much more pixels belong to the background (i.e. healthy tissue), so it is more likely that tumour pixels are overlooked.

6 Results, Comparison and Discussion

In fig. 5 and 6, an example for the segmentation of test image 84 is shown together with the box plots for the Dice score, sensitivity and specificity. We obtain similar scores for both techniques. The mean scores of the random forest are Dice$= 65.9\%$, sensitivity$= 77.1\%$ and specificity$= 99.1\%$, the mean scores of the U-Net are slightly higher with Dice $= 69.3\%$ and specificity $= 99.4\%$, the sensitivity$= 73.1\%$ is slightly lower.
The scores are also listed again in table 1, together with the scores of the other methods mentioned in section 2. Compared to the methods Maier et al. (2015), Dong et al. (2017) and Kamnitsas et al. (2017), our average performance is significantly worse. It needs to be noted however that the techniques were used for multi-class segmentation instead of binary segmentation, as we did in our case.

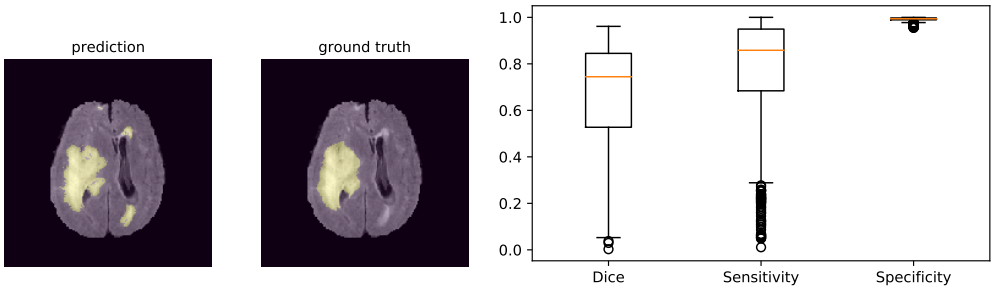


Figure 5: Results of the predictions using our random forest. *Left:* Example for a prediction on image 84 of the test set. The true tumour is segmented very precisely, but other smaller bright areas are also incorrectly classified as being a tumour. *Right:* Boxplots for the Dice score, sensitivity and specificity. As most of the images are background, the specificity is very high and has a low standard deviation. The quantiles Q_1 (25th percentile) to Q_3 (75th percentile) span almost over the whole range of the Dice score and sensitivity. This means that for a single image, the classification can vary significantly.

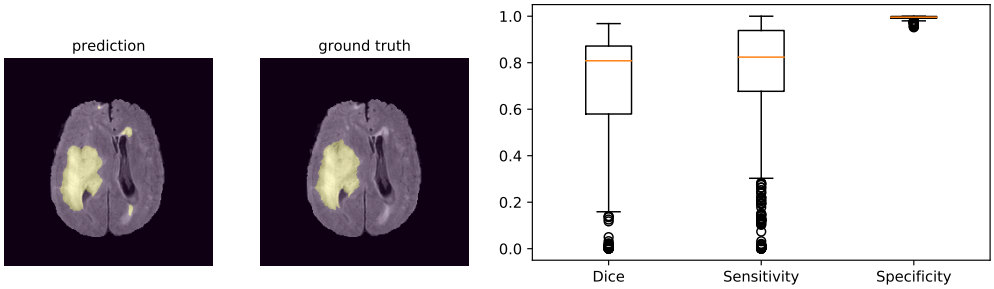


Figure 6: Results of the prediction using our U-Net. *Left:* Example for a prediction on image 84 of the test set. The segmentation of the true tumour is as accurate as for the random forest. The edges of the segmented tumour is smoother that compared to in fig. 5. The pixels incorrectly classified as tumour pixels, i.e. false positives, are also present in areas with high intensity, but smaller than for the random forest. *Right:* Boxplots for the Dice score, sensitivity and specificity. Just as for the random forest, the specificity is very high and has a low standard deviation. The median Dice score is slightly higher at about 0.8, the sensitivity score is however slightly lower as in fig. 5. The outliers all lie below the median, which is why the mean of the Dice and sensitivity scores is in fact lower.

The cases of our test set where the U-Net performed the worst and best with regard to the Dice score is shown in fig. 7. The figure also illustrates the strength and weaknesses of the U-Net. In both cases most of the tumour area is labelled as non-enhancing tumour in the ground truth. In the case with the lowest Dice score (top row) the U-Net is not able to detect the upper half of the tumour, but marks it as background (i.e. false negative). The intensity of the tumour appears similar to the normal tissue in the FLAIR image, only the edge has a higher intensity. In the case with the highest Dice score (top row), the intensity of the FLAIR image is visibly higher in the tumour region. The tumour area is

	Dice [%]	Sensitivity [%]	Specificity [%]
Random Forest	65.9	77.1	99.1
U-Net	69.3	73.1	99.4
Maier et al. (BRATS 2015, LGG)	84	85	-
Dong et al. (BRATS 2015, LGG)	84	-	-
Kamnitsas et al. (BRATS 2017, winner)	90.1	89.5	-

Table 1: Comparision of Dice score, sensitivity and specificity of the segmentation of the whole tumour between our random forest and our U-Net. The scores of methods by Maier et al. (2015), Dong et al. (2017) and Kamnitsas et al. (2017) are given as a reference.

detected correctly by the U-Net.
This shows that the U-Net is highly sensitive to areas with high intensity, but is not able to detect tumours when the intensity difference is small.

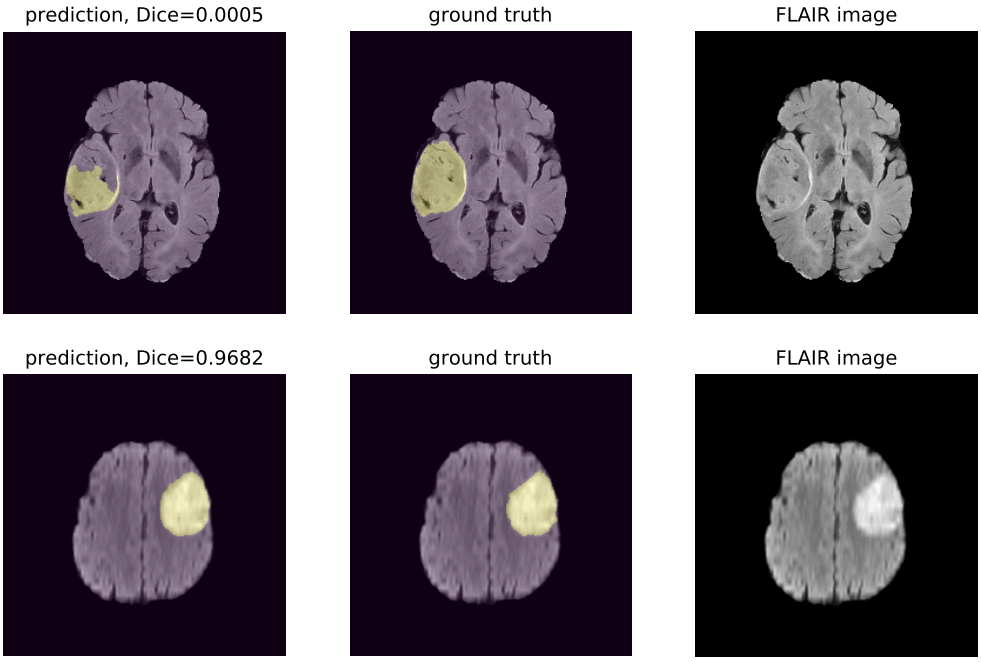


Figure 7: Predictions with our U-Net. *Top:* Prediction with overall lowest Dice score. *Bottom:* Prediction with overall highest Dice score.

7 Future Work

Our implementations only dealt with binary segmentation and were only possible to discriminate between tumour/ non-tumour regions. However, there is a larger variation in appearance from class to class, which could be addressed by training our methods on the classes separately.
In order to put more emphasis on "harder" examples, which are mainly tumours with a lower intensity contrast, the each training image could be weighted accordingly.
The shorter training time of the U-Net leaves room to increase the training data set by for example also include HGG cases or data augmentation.
Finally, another possibility is to combine both methods as ensemble and use features from the U-Net for the random forest, similar to the work by Kamnitsas et al..

8 Conclusion

In this project, we tested two approaches to perform tumour segmentation on MRI images, a random forest and a U-Net.
Even though our techniques were less accurate than other state-of-the-art techniques, we were able to implement, train and test them in a reasonable amount of time, when considering our limited resources.
Both techniques lead to similar segmentation accuracies. Training and inference time with the U-Net are significantly smaller and it is expected to be able to increase the segmentation accuracy by a deeper U-Net, longer training time, more training data and hyperparameter tuning. To increase the prediction accuracy of the random forest, new features have to be introduced. As it is difficult to decide which features will lead to better results, improving the U-Net is easier. Also, adding more training data does only lead to a longer training time for the U-Net, but for the random forest it additionally requires more memory resources, making this method less flexible.

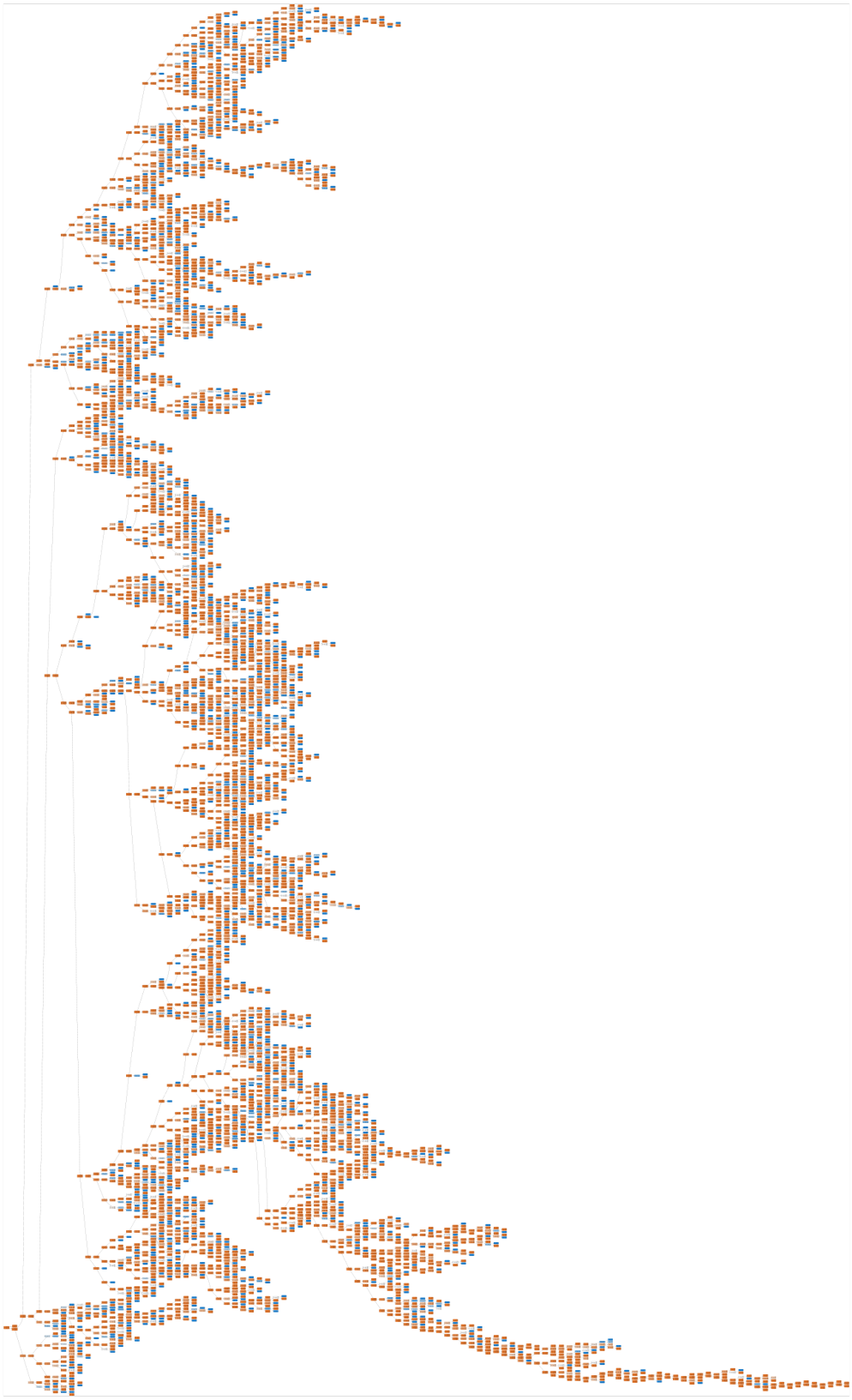
9 Appendix

9.1 Hardware Specifications

This section lists the hardware details of the machine used to train the random forest and the U-Net. This makes the given times more comparable to other systems.

- Intel Core i5-3550 CPU 3.30GHz
- Nvidia Geforce GTX 1060 6GB
- 8 GB RAM
- 40 GB swap on SSD

9.2 Visualization of a single tree



References

Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4: 170117, 2017.

Zijun Deng. Pytorch for semantic segmentation. `https://github.com/zijundeng/pytorch-semantic-segmentation`, 2017.

Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *Annual Conference on Medical Image Understanding and Analysis*, pages 506–517. Springer, 2017.

Tin Kam Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

Jackson Huang. U-net implementation in pytorch. `https://github.com/jaxony/unet-pytorch`, 2017.

Konstantinos Kamnitsas, Wenjia Bai, Enzo Ferrante, Steven G. McDonagh, Matthew Sinclair, Nick Pawlowski, Martin Rajchl, Matthew C. H. Lee, Bernhard Kainz, Daniel Rueckert, and Ben Glocker. Ensembles of multiple models and architectures for robust brain tumour segmentation. *CoRR*, abs/1711.01468, 2017. URL `http://arxiv.org/abs/1711.01468`.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. URL `http://arxiv.org/abs/1411.4038`.

Oskar Maier, Matthias Wilms, and Handels. Highly discriminative features for glioma segmentation in mr volumes with random forests. In *Proceedings of the Multimodal Brain Tumor Image Segmentation Challenge*, pages 42–45, 2015.

B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, Oct 2015. ISSN 0278-0062. doi: 10.1109/TMI.2014.2377694.

Quinn T Ostrom, Luc Bauchet, Faith G Davis, Isabelle Deltour, James L Fisher, Chelsea Eastman Langer, Melike Pekmezci, Judith A Schwartzbaum, Michelle C Turner, Kyle M Walsh, et al. The epidemiology of glioma in adults: a "state of the science" review. *Neuro-oncology*, 16(7):896–913, 2014.

Adriano Pinto, Sérgio Pereira, Higino Correia, Jorge Oliveira, Deolinda MLD Rasteiro, and Carlos A Silva. Brain tumour segmentation based on extremely randomized forest with high-level features. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 3037–3040. IEEE, 2015.

M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool. Incremental learning of random forests for large-scale image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):490–503, March 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2459678.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.