

# **Semantic Brain Tumor Segmentation Using Conditional Adversarial Networks**

**Seminar  
Practical Video Analysis  
Summer Term 2017**

Konstantin Harmuth,  
Willi Gierke,  
Thomas Kellermeier,  
Martin Fischer

Supervisors:  
Mina Rezaei  
Dr. Haojin Yang  
Prof. Dr. Christoph Meinel

October 3, 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Related Work</b>	<b>3</b>
<b>3</b>	<b>Dataset</b>	<b>5</b>
<b>4</b>	<b>Tumor Classification</b>	<b>6</b>
4.1	Healthy vs. Unhealthy . . . . .	7
4.2	Low-Grade Glioma vs. High-Grade Glioma . . . . .	7
<b>5</b>	<b>Tumor Segmentation</b>	<b>8</b>
5.1	Conditional Adversarial Networks . . . . .	8
5.2	Network Architecture . . . . .	10
<b>6</b>	<b>Survival Rate Prediction</b>	<b>11</b>
6.1	Using Support Vector Regression . . . . .	12
6.2	Using Convolutional Neural Networks . . . . .	12
6.2.1	Model . . . . .	14
6.2.2	Results . . . . .	14
<b>7</b>	<b>Demo Tool</b>	<b>16</b>
<b>8</b>	<b>Conclusion and Future Work</b>	<b>17</b>
8.1	Segmentation . . . . .	17
8.2	Segmentation Improvements . . . . .	17
8.3	Survival Rate Prediction . . . . .	18
	<b>References</b>	<b>19</b>

## 1 Introduction

Gliomas are the most common types of brain tumors. They vary in aggressiveness, sub-regions and potential treatment which is reflected in a very diverse appearance. This makes segmenting gliomas in multimodal scans a greatly challenging task.

The Brain Tumor Segmentation Challenge (BraTS challenge)<sup>1</sup> encourages researchers to develop algorithms to support doctors in segmenting brain tumors by providing a large dataset of brain scans of patients who suffered of gliomas. Each patient's brain was scanned using four different image modalities. Additionally, a set of expert neuroradiologists labeled each pixel of a 3D scan with the information whether the pixel belongs to the healthy brain or to one of three parts of the tumor. The goal of the challenge is to support the treatment of high-grade (HGG) and low-grade gliomas (LGG) by evaluating the progression of the disease and the success of the chosen treatment strategy.

Since renting hospital machines to perform scans, annotating them manually and getting the legal clearance to use that data of people is extremely expensive and cumbersome, there are only scans of 285 patients in the dataset. Since this is a very small amount of data, we used an adversarial network that creates its own training data during training time to improve itself.

This report is structured as follows. First, Section 2 gives an overview of algorithms fellow researchers used to segment objects. Secondly, Section 3 shows the attributes of the used dataset. The next Section 4 explains how we distinguished different brain and glioma types from each other to get a feeling for the topic. Next, Section 5 presents the network we used to solve the BraTS challenge with outstanding performance in comparison to the other participants. Section 6 introduces the approaches we used to solve the second task of the BraTS challenge, which was to predict the overall survival rate of a patient. In Section 7 we give an overview of the on-line tool we built that is based on the segmentation algorithm from Section 5 to segment tumors in brain slices using a web service. Lastly, Section 8 completes this report by discussing our findings and suggesting further improvements and extensions.

## 2 Related Work

Semantic image segmentation is an active topic of research, fanned by various challenges. Therefore a wide range of models exists that attempt to solve this problem.

One common approach for segmentation is the U-Net [1]. It uses a fully-convolutional architecture consisting of an encoder and a decoder. The encoder is able to capture contextual information while the decoder enables precise localization. Furthermore, skipped connections between symmetric layers enable important information to directly flow from the encoder to the decoder. This helps to avoid the vanishing gradient problem since the gradient from the top layer will certainly reach the bottom layer. However, this

---

<sup>1</sup><http://www.med.upenn.edu/sbia/BraTS2017.html>

model still consists of a large amount of parameters which results in extensive memory consumption.

A simplified approach is the SegNet [2], which reuses pooling indices instead of transferring the whole feature map to the corresponding decoders like U-Net does. The authors of SegNet also propose to use pre-trained convolutional layer weights from the VGG net as pre-trained weights, accelerating the training time. On the one hand, SegNet needs less memory due to reusing the pooling indices. On the other hand, it was designed to understand road and indoor scenes that typically consist of more different objects than there are in medical image scans. Thus, it is questionable whether the approach is suitable for our goal.

A different way of handling pixel-wise segmentation shows DilatedNet [3]. To access local and global image information as well it uses dilated convolutions. Dilated convolutions basically skip pixels, which enables the layer to increase the receptive view exponentially while the number of parameters grows linearly. Figure 1 explains this approach visually. While the concept of skipping pixels to reduce memory consumption is suitable for detecting many objects in the same image, it's questionable whether it's also able to detect medical objects in highly unbalanced data sets we are confronted with.

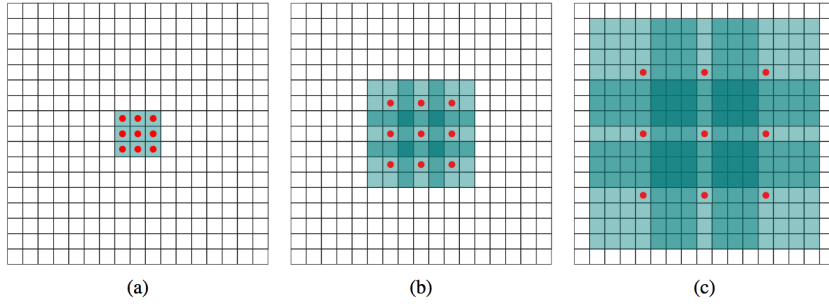
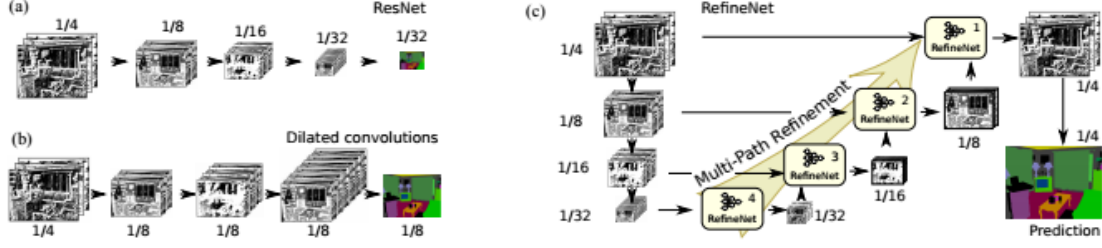


Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a)  $F_1$  is produced from  $F_0$  by a 1-dilated convolution; each element in  $F_1$  has a receptive field of  $3 \times 3$ . (b)  $F_2$  is produced from  $F_1$  by a 2-dilated convolution; each element in  $F_2$  has a receptive field of  $7 \times 7$ . (c)  $F_3$  is produced from  $F_2$  by a 4-dilated convolution; each element in  $F_3$  has a receptive field of  $15 \times 15$ . The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

**Fig. 1:** The principle of dilated convolutions explained by Yu and Koltun[3].

RefineNet [4] is a fully convolutional neural network that combines feature maps at different stages of convolutions to achieve high-resolution segmentations in an economic way. A comparison of the approaches of fully convolutional neural networks, dilated convolutions and RefineNet is shown in Figure 2.

All of these approaches for segmentation work on two-dimensional images and only process one image at a time. Medical data, like given by the BraTS challenge, is often available as three-dimensional images because it was recorded using magnetic resonance imaging (MRI) or X-ray computed tomography (CT). For this reason there are also



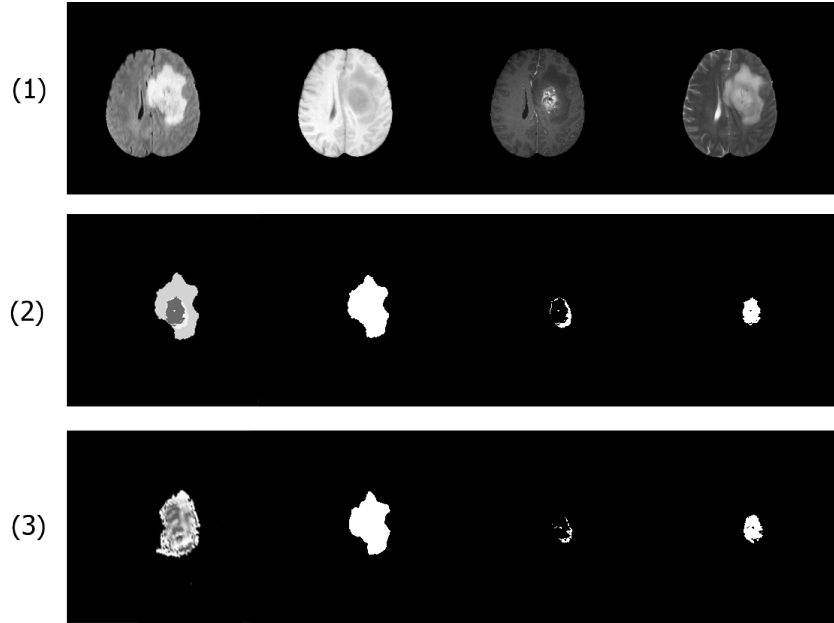
**Fig. 2:** Standard CNNs save memory while achieving only low resolutions. Dilated convolutions (like they are used in SegNet) can accomplish higher resolutions than vanilla CNNs at the expense of computation power and memory consumption. RefineNet claims to be able to achieve even better resolution while saving both computation power and memory by merging different detail levels of feature maps throughout various convolution stages.

some models that work with three-dimensional data. This is necessary because the segmentation problem is sometimes too difficult to solve only using two-dimensional data. Cascaded FCN[5] and V-Net[6] both adapt the U-Net to work with volumetric data for medical image segmentation while being very demanding regarding GPU memory and needed training data.

### 3 Dataset

This section describes the used data to complete the BraTS challenge. There are two types of data used:

1. **Brain image data:** Images of a brain, recorded using CT. The data is available in three orthogonal axes. For each axis there are four different modalities that differ in the visible features as you can see in Figure 3. For each image the non-brain parts of the data is removed. The images are available in gray-scale and have a size of 240x240 pixels. One pixel equals 1mm in the real world.
2. **Survival data:** The age and the number of days the patient died after the image was made. Each patient could be identified by an ID so we were able to match the data for task two with the data and results for task one. This data is not available for every patient.



**Fig. 3:** (1) Input images containing the FLAIR, T1, T1CE and T2 scans (from left to right) of the same brain slice of a patient. The non brain parts of the image were removed.  
(2) Ground truth of the tumor (left). The BraTS challenge requires the segmentation of three different tumor regions which can be seen as binary masks (right).  
(3) Prediction of the pix2pix model (left) that was trained on the ground truth and our improved model (right) that was trained using the binary masks.

## 4 Tumor Classification

We started with a simple task to get to know the data. At first a two-dimensional slice of a patient's brain was analyzed to find out if there is any brain tumor. If the patient is not suffering from a tumor, the task is fulfilled. In case there is a tumor, it's important to know how dangerous it is. Therefore the tumors have to be categorized into HGG and LGG. This is important for the second task when the survival rate of a patient will be inspected and predicted. Those two classification tasks weren't part of the BraTS challenge this year, but in a real scenario it's a useful information.

To provide enough data to train on we decided to split up each patient into single images. After removing blank images 15.000 images remained. Each one of these images was considered independently of the others so there were enough datapoints to train on.

### 4.1 Healthy vs. Unhealthy

For this simple classification task a small convolutional neural network (CNN) was used containing some convolutional, max pooling, activation (using ReLU), dropout, flatten and dense layers as shown in Figure 4. We trained four different networks with almost the same structure. Three of them got only images regarding one of the three available axes. The last one was trained on all axes so it could handle all types of views.

Surprisingly the network was always able to reach 100 % accuracy on the validation set after one epoch of training with about two third of the provided data. It was easy to train on the provided data because healthy and unhealthy images had some magnificent differences.

In our first approach the images containing a tumor had no skull and were brighter than the healthy image. The second dataset was augmented so the unhealthy images also contained a skull. Unfortunately, one could clearly see the difference between augmented and real brain scans. Examples of the first and second datasets for healthy and unhealthy patients are shown in Figure 5. Again it was very easy for the network to converge. After scaling the images to a low resolution the calculations were very fast and initially produced better accuracy before converging to the optimum.

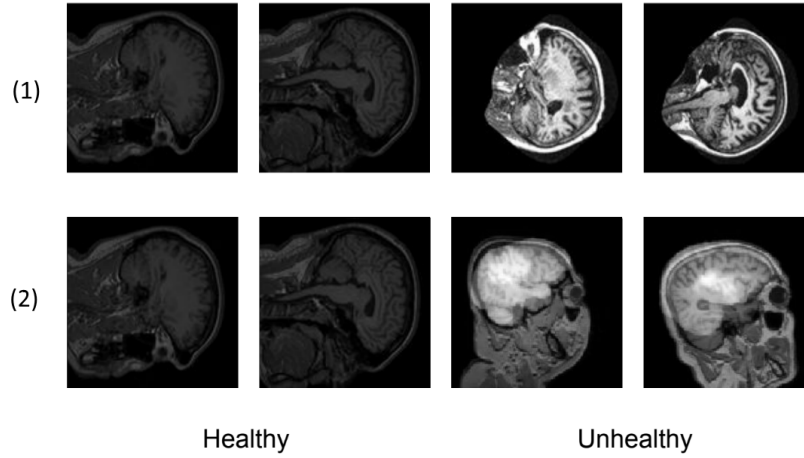
Conv2D
MaxPooling2D
Conv2D
MaxPooling2D
Dropout
Flatten
Dense
Dropout
Dense

**Fig. 4:** CNN classifying patient's healthiness and the type of glioma

### 4.2 Low-Grade Glioma vs. High-Grade Glioma

For the treatment of the tumor and the patient's survival rate it's important to know which type of tumor the patient is suffering from. As pointed out by Menze et al. [7] in their paper related to the BraTS challenge the life expectancy highly varies depending on the type of glioma. Usually patients will survive several years if they're suffering from a LGG. In contrast to this, patients with the more aggressive HGG have a median survival rate of two years or less.

For this task we were looking at the ground truth dataset for task one which solely consists of images of the tumor itself instead of the whole brain. Again as in the previous classification task there were only two classes and the network received a similar input shape. That's why we decided to apply the same model on this task. The network was trained on another dataset containing 25.000 tumor images and again converged very quickly with about 95 % accuracy as you can see in Figure 6.



**Fig. 5:** (1) Comparison of four patients, two of whom suffering from a tumor. Notice the different brightness and the missing skull on the scans of the unhealthy patients. (2) The unhealthy labeled scans from row (1) are augmented so they contain a skull. The difference in brightness still remains.

## 5 Tumor Segmentation

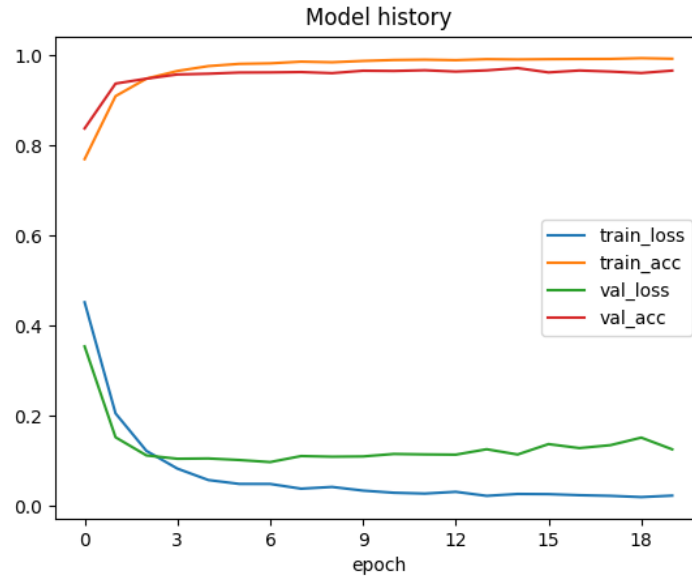
### 5.1 Conditional Adversarial Networks

As we already stated in Section 4, having too few data is a giant problem when trying to apply Deep Learning algorithms. Even when trying to artificially extend the available data by e.g. adding tumors to known positions of healthy scans, it is easy to introduce properties to the images that make it easier for the applied algorithms to find those artificial tumors than real tumors. To solve this problem, we used a Conditional Generative Adversarial Network (cGAN) for the tumor segmentation, which is a special kind of Generative Adversarial Network.

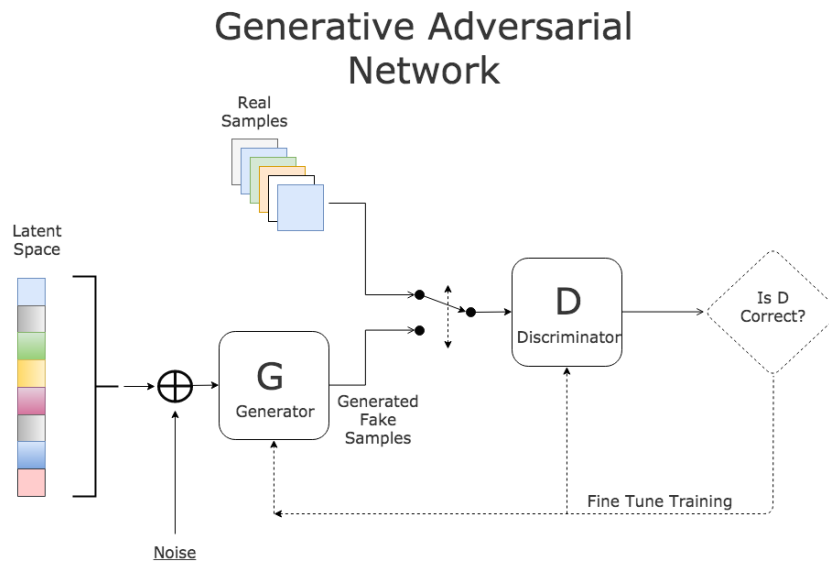
Generative Adversarial Networks (GANs) have been introduced by Goodfellow et al.[8]. They learn to artificially create objects that highly resemble samples of a given real dataset. A simplified depiction of a GAN can be found in Figure 7.

They consist of two agents: a discriminator and a generator. The generator receives a vector containing random noise as input and generates an output. The discriminator receives this output of the generator and a sample of the real dataset and decides, whether the given input is real or fake, meaning its origin is the dataset or the generator, respectively. The discriminator receives feedback whether its guess was right or wrong. The generator in turn receives feedback whether he was able to "fool" the discriminator, meaning the generated object resembled the samples of the real dataset so well that the discriminator was not able to distinguish between them.



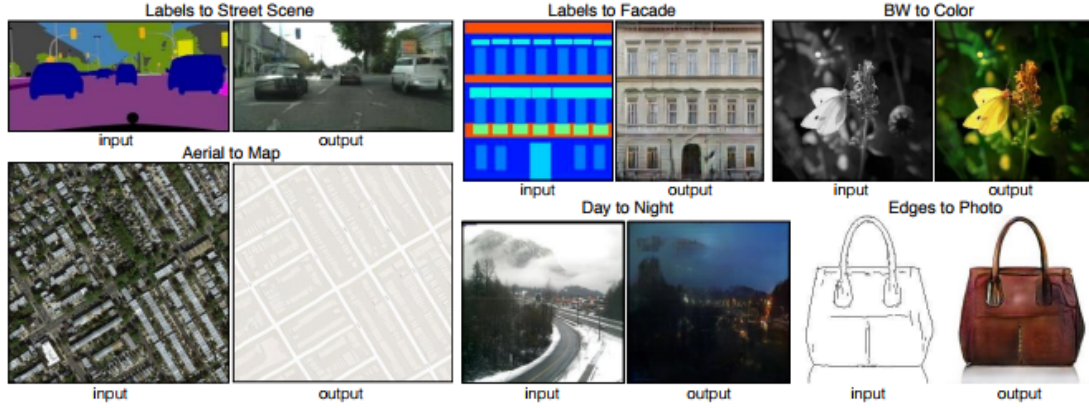


**Fig. 6:** Using a very similar network as in the previous classification task we got a high accuracy on classifying HGG and LGG.



**Fig. 7:** A GAN that learns to generate fake samples that can hardly be distinguished from real samples (from [9])

Conditional Generative Adversarial Networks cGANs have been introduced by Isola et al.[10].



**Fig. 8:** Isola et al. show that cGANs are very well suited for image-to-image translations like colorizing black and white pictures or transforming day pictures to night pictures.

The real dataset of cGANs consists of picture tuples of an input and an output image. Additionally to random noise, generator of a cGAN also receives the input image and is supposed to generate an output image that is accepted by the discriminator. The discriminator, on the other hand, learns to decide whether an output image fits to the given input image or not. Thus, the generator is more restricted to the mapping he is supposed to learn. Isola et al. were able to show that this creates very promising results even with very complex graphical structures.

## 5.2 Network Architecture

Our segmentation network is based on pix2pix<sup>2</sup>, which is a cGAN. This model uses U-Net [1] for the generator and a Markovian Discriminator [10]. The network produces good results with generative tasks. Our contribution is to test the applicability for medical image segmentation. To achieve this it was necessary to refactor the implementation, albeit the model structure was not changed.

The most important change was made regarding the used data. The original implementation used a RGB image as an input and predicted a RGB image as well. A first test with this model produced very bad results (Figure 3). The reason for this is mainly restrictions regarding the input and the output of the model. It only supports one image for both, but to get good results we need all four modalities (Figure 3). However the biggest improvement was to split the ground truth into three different label masks. Each

<sup>2</sup><https://github.com/yenchinlin/pix2pix-tensorflow>

mask encodes only one label. This eliminates the gradients between different labels in one image. To retrieve the actual output it is required to merge the different label masks again.

To improve the model performance we use gray-scale images instead of RGB images. This reduces the memory consumption to a level that is comparable with the original implementation. This leaves room for further expansion of the used model. To improve the generator we doubled the depth of each convolutions and also increased the size of each convolution from 5x5 to 6x6. Both changes increased the accuracy at the cost of a much longer training time.

Another way to improve the result is by using smart data selection. Instead of training the model with images from every axis, we only use one axis. This increases the result for one axis drastically, because each axis contains a different set of features. Also the size of the images is not uniform for all axes. To reduce the training time we additionally implemented virtual batch normalization [11]. To achieve a faster training process, we ordered the data, so that each batch only consisted of similar images.

## 6 Survival Rate Prediction

The consecutive second task of the BraTS challenge was to use the segmented tumor of the patient to predict the patient's overall survival rate. The only additional feature that was given was the patient's age. The label was a number between zero and 1750 which represented the number of days the patient lived after the scans were taken. Thus, this task might have been even more challenging than the first one because the information-to-noise ratio was even worse as the target label was only a single number for a whole 3D brain scan that included segmented tumors. Additionally, any wrong segmentation that was made during task one would also influence this task's prediction. To artificially create more training data, we considered using data augmentation. Though, while we're no experts in the field of neuroscience, we had to assume that the size and location of a tumor highly influences the patient's survival rate. Thus, we rejected using data augmentation by scaling, rotating or moving the segmented tumors as it could have falsified the survival rate, introducing even more noise into the dataset.

For evaluating the importance of the patient's age we looked at the correlation of the age and the survival rate. As you can see in in Figure 9 the survival rate slightly decreases with an increasing age of the patient.

To predict the survival rate of a patient, one team member tried to use classical machine learning models while another one used a convolutional neural network approach. The information regarding brain and tumor of a patient was given as seven 3D images. Three of them were representing the tumor regions while the remaining four were the input for task one showing the brain using different modalities. Each 3D model had a shape of 120 Pixels x 120 Pixels x 129 Slices. To break down the tumor segments

to comparable variables, we did some analysis and extracted values like the size of the tumor:

- **TumorSum1:** All pixels labeled as part of the first tumor region
- **TumorSum2:** All pixels labeled as part of the second tumor region
- **TumorSum3:** All pixels labeled as part of the third tumor region
- **TumorSum:** Sum of the previous three values
- **BrainSum:** All pixels labeled as part of the brain

We used the sum of pixels to represent the size of a 3D object. Because all images had the same resolution, it would have ended up in scaling the values by the same factor so the correlation matrix would look the same.

Figure 10 shows the correlation matrix for the patient's age, the survival rate and the other five previous mentioned variables. Although the correlation between age and survival rate seemed to be very weak regarding Figure 9 it is the strongest one among all the other variables in comparison with the survival rate.

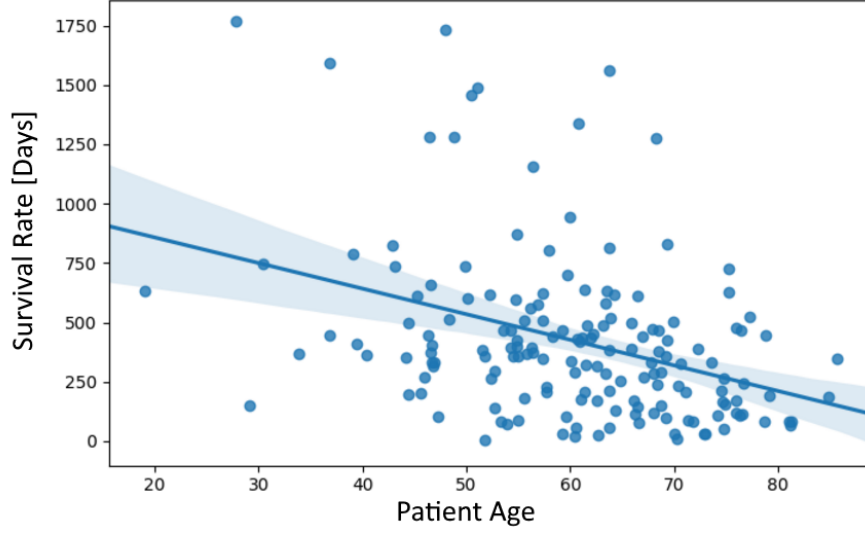
### 6.1 Using Support Vector Regression

Using the Python library scikit-learn[12], we collected all regression models (Bayesian Ridge Regression, Stochastic Gradient Descent Regressor, Support Vector Regression etc.) and cross-validated them on the aforementioned features of the training data.

The Support Vector Regression (SVR) clearly outperformed the remaining regression approaches. To fine tune the model, we tried different kernels that are mapped on the data before the regression function is fit to the data. In Figure11 one can see that the SVR with the lowest mean squared error (MSR) basically just learned to predict the average of the survival rate distribution. This is a strong indicator that the approach is still in the early stages of development.

### 6.2 Using Convolutional Neural Networks

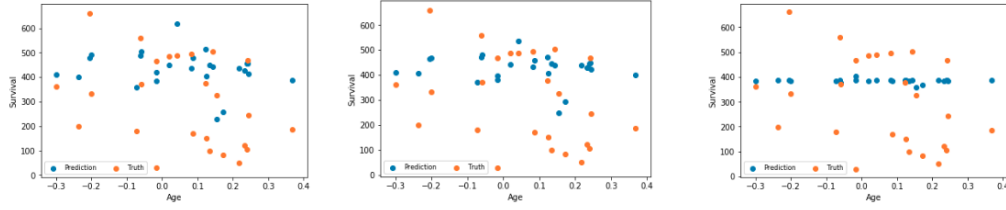
In contrast to the approach of using Support Vector Regression we added the segmented tumor from task one. The image data provides information regarding the position and the size of the tumor which might influence the survival rate of the patient.



**Fig. 9:** The survival rate of a patient decreases with an higher age of the patient. The datapoints are widely spread so we have a weak correlation. The Pearson's correlation coefficient overall 163 datapoints is about  $-0.37$ .

	Age	Survival	TumorSum	TumorSum1	TumorSum2	TumorSum3	BrainSum
Age	1.000000	-0.372145	0.016000	0.043340	0.016704	-0.098971	-0.123452
Survival	-0.372145	1.000000	-0.158764	-0.118707	-0.180738	-0.164048	0.033865
TumorSum	0.016000	-0.158764	1.000000	0.944752	0.913304	0.740194	0.212170
TumorSum1	0.043340	-0.118707	0.944752	1.000000	0.747455	0.524778	0.242744
TumorSum2	0.016704	-0.180738	0.913304	0.747455	1.000000	0.779349	0.145269
TumorSum3	-0.098971	-0.164048	0.740194	0.524778	0.779349	1.000000	0.094524
BrainSum	-0.123452	0.033865	0.212170	0.242744	0.145269	0.094524	1.000000

**Fig. 10:** The correlation between all extracted numerical values regarding the second task. Note that the size of a brain decreases with the patient's age as well as the patient's survival rate.



**Fig. 11:** We varied the kernels that were applied to the data, namely a radial basis function, a sigmoid and a quadratic kernel (from left to right). The values for the MSR were 44.000, 43.000 and 39.000. It's worth noting that the smaller the error gets, the more often the related SVR predicts the average value of the survival day distribution.

### 6.2.1 Model

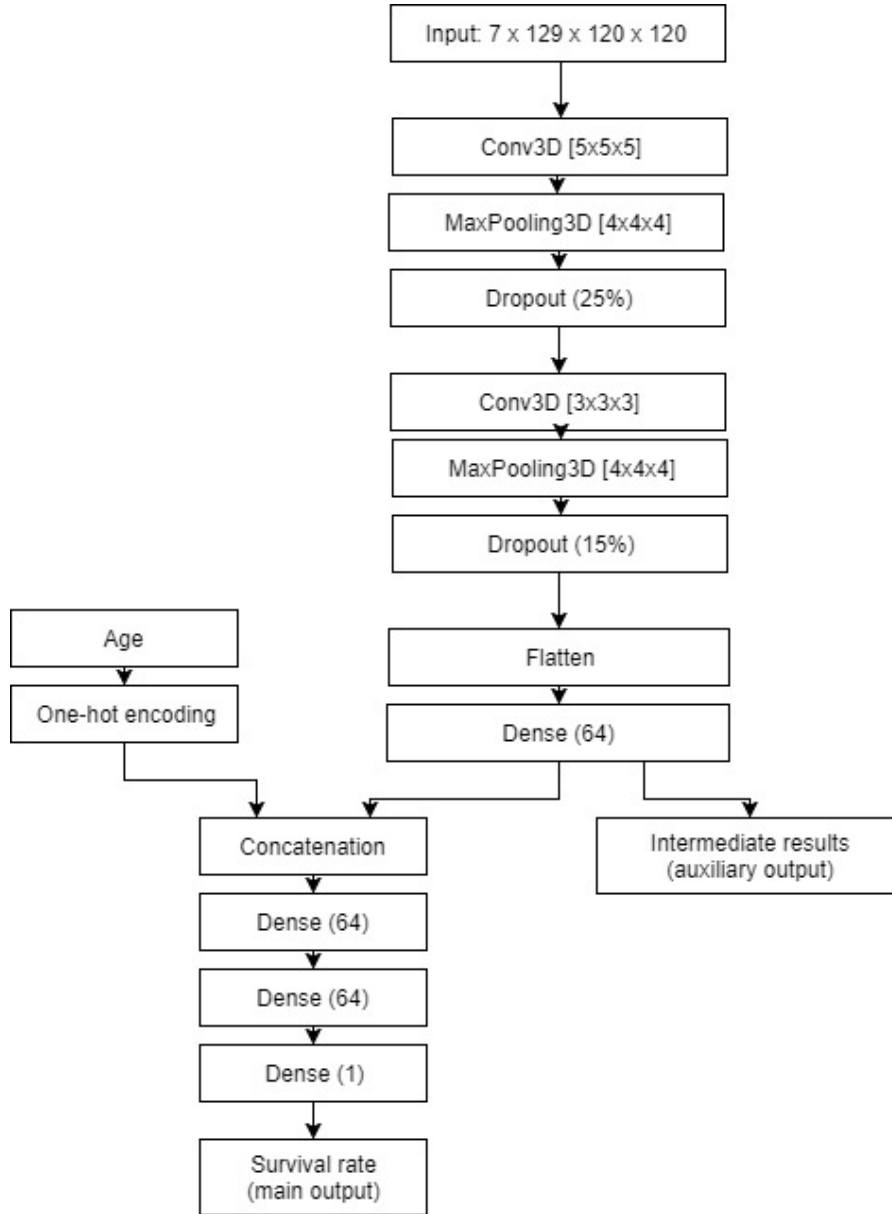
As previously mentioned we assumed that the tumor's position might be important for the survival rate. That's why we decided to use a CNN respecting the whole brain and the tumor instead of extracting usable information as done in our previous analysis.

Our network processing the different types of input data is shown in Figure 12. To merge the multi-input including different data types we used a CNN which is combined with the patients age via a concatenation layer. In the end we applied some final dense layers ending up in a number representing the survival rate of the patient.

The age of patient is a number within the interval from 0 to 100. Because we expect a non linear behavior regarding the survival rate we wanted to transform the age into classes. This way each class can influence the whole regression task in another way and the input is limited to a certain range of classes. The transformation is done by applying one-hot encoding which is a usual procedure in the field of machine learning to transform some values into classes.

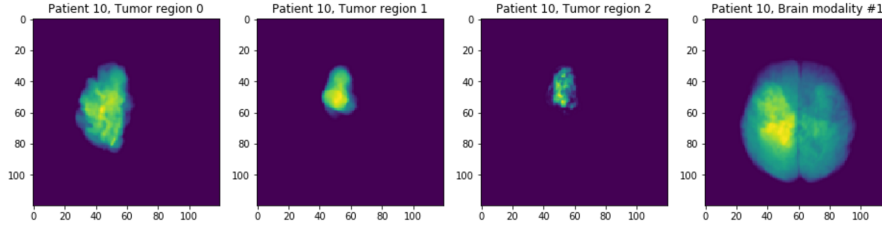
### 6.2.2 Results

Because we're missing the most important part for our work, which is a lot of data, our network didn't converge to any reasonable results. We tried out different optimizer, decreased the dropout rate, reduced the input size in different ways and applied many other changes on our network structure, but unfortunately we couldn't fix this problem. One of our attempts included transforming the 3D models into 2D images. Instead of considering the three-dimensional arrangement within the brain we look at a cross sectional area. This cross-section image visualized the density of a two-dimensional point over the whole third dimension. An example for the output of this transformation is shown in Figure 13.



**Fig. 12:** Rough structure of the network receiving a patient's age and images regarding his tumor and his whole brain. It's based on the concatenation network from the Keras documentation <sup>3</sup>. The main output is a float number between zero and one which gets scaled up to 2000.

<sup>3</sup> [https://keras.io/getting-started/functional-api-guide/](https://keras.io/getting-started/functional-api-guide/#multi-input-and-multi-output-models)  
#multi-input-and-multi-output-models

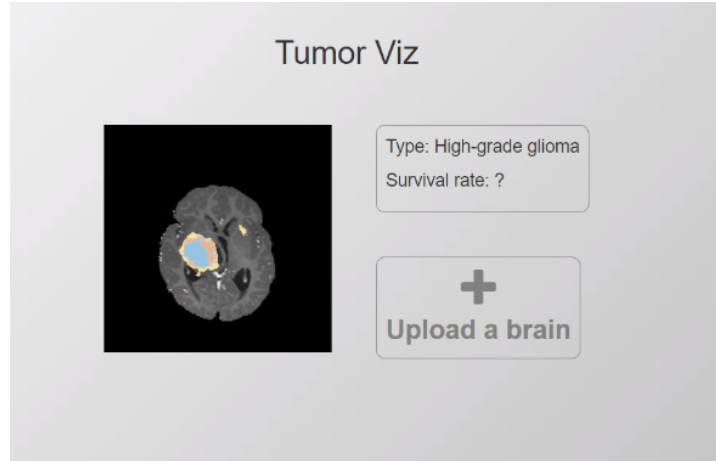


**Fig. 13:** Outcome of the transformation from 3D to 2D images for three tumor regions and one of the four given brain modalities.

## 7 Demo Tool

Based on our model that has been developed in Section 5, we built an interactive demo tool. As an input, it takes a PNG image file consisting of the FLAIR, T1, T1CE and T2 scans of a patient's brain slice. For this example we consider as input the picture depicted in Figure 3. The server needs a few seconds to compute the predicted tumor classes, given there are any. The output image can be seen in Figure 14. The tool highlights the necrotic and non-enhancing tumor in blue, the GD-enhancing tumor in red and the peritumoral edema in yellow. In this example, the server recognized the tumor as a HGG.

For the direct usage of our trained networks we've set up a python server which receives the images of a patient and returns the results of our networks. Our frontend application is based on JavaScript using Redux and React.



**Fig. 14:** Demo tool with segmented tumor and classified tumor type based on input image from Figure 3



## 8 Conclusion and Future Work

This section contains a conclusion for both BraTS challenges.

### 8.1 Segmentation

The segmentation model we proposed is successfully predicting tumor positions. We definitely improved the approach we worked on as one can see in Figure 3 (3). However, our prediction is still far away from being perfect which has multiple reasons. First of all we only use two-dimensional images and do not combine features of neighbouring images. This leads to missing very small tumors and sometimes predicting very small, false tumors at random locations. One possibility to solve this problem is by using the three-dimensional information that is present in the data. Missing small tumors decreased our rank in the BraTS challenge a lot because for each patient the accuracy was averaged. We usually achieved accuracies above 90% but the false positive detections were calculated as 0% and impaired our final rank.

While working on our model we found some additional flaws in it. The biggest mistake is probably the choice of the loss function. We use multiple masks with sigmoid-cross-entropy but every other segmentation model uses a softmax layer with categorical-cross-entropy. Our approach has the drawback that we needed to merge the three tumor masks we predicted in order to submit our result. A problem with this operation is that there is no good way of merging overlapping masks. Having one softmax layer for each pixel would solve this problem because we would only generate one label per pixel. This would also solve the problem that we do not scale well for many different segmentation classes.

Another problem is that our mask method requires post-processing to generate the output. Due to not using softmax we always generate images that contain gradients between -1 and 1 which have to be mapped to binary values using the sigmoid function. The three binary masks then have to be merged into one image.

Due to the use of U-Nets our model has lots of parameters so we need a long training time (basically two weeks on two Titan X GPUs). The segmentation performance is about one fps.

To keep the U-Net encoder/decoder simple we also had to always use images with dimensions of a power of two. Since the shapes of the images of the BraTS data are not a power of two, we always had to pad the scans. The perceived flaws are leaving enough space for further improvements.

### 8.2 Segmentation Improvements

One way to improve the results is to use a refinement network to post-process to generated predictions. This network could use the sequence property of the used medical data. To use this additional information it is possible to use a U-Net that utilizes a

LSTM node to process the encoded input. This model should learn a different set of features than the first one. In order to achieve this we would not train on the ground truth. Instead we would train using the error of the original prediction. The model then learns an error function that can be used to further improve our results.

The LSTM idea can also be directly implemented into the cGAN. This way the generator and discriminator themselves are aware of the sequence property. A problem with this approach is that it reduces the amount of input data from 285x155 images to 285 sequences, which increases the problem of overfitting.

### 8.3 Survival Rate Prediction

While we can detect the presence and the type of a brain tumor and also received good results in segmenting the tumor region, we were not able to predict the patient's survival rate in a satisfying matter. In comparison to the other tasks we couldn't augment data to create a bigger training dataset. Furthermore, it included a more complex relation between input and output. We inspected the correlations between the received data and the ground truth labels to get a better understanding of the correlations. Unfortunately our assumptions were not precise enough to successfully train a network on the dataset. The survival rate might depend on several other factors than the patient's age so it would be helpful to have more medical research on this topic to improve the dataset.

## References

- [1] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” CoRR, vol. abs/1505.04597, 2015.
- [2] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” arXiv preprint arXiv:1505.07293, 2015.
- [3] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” CoRR, vol. abs/1511.07122, 2015.
- [4] G. Lin, A. Milan, C. Shen, and I. D. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation,” CoRR, vol. abs/1611.06612, 2016.
- [5] P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, M. Armbruster, F. Hofmann, M. D’Anastasi, W. H. Sommer, S. Ahmadi, and B. H. Menze, “Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3d conditional random fields,” CoRR, vol. abs/1610.02177, 2016.
- [6] F. Milletari, N. Navab, and S. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” CoRR, vol. abs/1606.04797, 2016.
- [7] 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. R. 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, 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. T. Ryan, D. Sarikaya, L. H. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Székely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. B. Ünal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. V. Leemput, “The multimodal brain tumor image segmentation benchmark (brats).,” IEEE Trans. Med. Imaging, vol. 34, no. 10, pp. 1993–2024, 2015.
- [8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems 27 (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

## References

---

- [9] KDnuggets<sup>TM</sup>, Business Analytics, Big Data, Data Mining, Data Science, and Machine Learning, “Generative adversarial networks – hot topic in machine learning,” 2017. [Online; accessed september 18, 2016].
- [10] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *CoRR*, vol. abs/1611.07004, 2016.
- [11] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” *CoRR*, vol. abs/1606.03498, 2016.
- [12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.