Volumetric Semantic Segmentation of Glioblastoma Tumors from MRI Studies

Ademi Adeniji
Stanford University
450 Serra Mall
ademi@stanford.edu

Vincent Liu
Stanford University
450 Serra Mall
vliu15@stanford.edu

Abstract. *Glioblastoma (GBM) is the most common and lethal primary, malignant, grade IV brain tumor in adults. Automated segmentation of GBM lesions from gadolinium-enhanced magnetic resonance imaging (MRI) is necessary to diagnose the presence of such tissue abnormality in the cranium. Human segmentations are expensive, time-inefficient, prone to error, and also require anatomical expertise. We seek to help automate this process by developing a semantic segmentation network for volumetric (3D) subregion segmentation of MRI images for Stanford Cancer Institute's larger subtype classification pipeline. The existing baseline implementation for this use case achieves a 60% dice coefficient score with a V-net [9] on the Stanford Cancer Institutes test data and over-segments on non-tumor parts of the brain such as the skull and orbits. In our approach, we adopt the commonly-used encoder-decoder convolutional architecture similar to the U-net [11]. Additionally, we leverage a variational autoencoder (VAE) branch to reconstruct the MRI input images, which imposes a regularizing constraint on the encoder [10]. The combination of both achieves a 64% dice coefficient score outperforming the baseline quantitatively and qualitatively.*

1. Introduction

Magnetic resonance images (MRI) serve to display the severity and location of tumor tissue which can determine subsequent surgical approaches. However, MRIs can also be used to sub-classify GBM tumor clusters for use in diagnosis. Researchers at the Stanford Cancer Institute identified 128 radiomic features [6] for classifying the rim-enhancing sub-type, which makes up 1/3 of all GBM lesions, leaving a high-precision lesion segmentation as one of the final pieces left to a complete lesion classification pipeline.

The input to our volumetric convolutional model is a set of modalities, or color contrasts, of skull-stripped MRI images of a single patient's brain. There are four major MRI modalities, or color contrasts, of a given MRI scan: t1, t2, t1ce (t1-post), and flair. We only utilize the t1-post and flair modalities since the Stanford Cancer Institute's test data is most complete for those. The output of the model is a volume of the same dimension of an input MRI image of a single modality, but with a multiclass label mask for the voxel values. The problem formulation is multiclass semantic segmentation, where the model learns to label each voxel with one of three classes corresponding to a given tumor subregion or no class, indicating healthy brain tissue. The three different tumor sub-regions are "enhancing tumor", "tumor core", and "whole tumor", which correspond to labels 4, 1, and 2 [1]. We opt to train the model on separate binary classifications instead of a single multiclass classification to simplify the objective in training. As is common in most semantic segmentation tasks, we seek to optimize the dice score coefficient, which evaluates the quality of the model's learned predictions.

2. Related Work

2.1. V-net architecture

Recent work in computer vision in the medical imaging space has demonstrated the effectiveness of encoder-decoder convolutional architectures in detection, classification, and segmentation tasks. The Stanford Cancer Institute's current segmentation model baseline is the V-net [9], which encodes the input volume through a series of 5x5x5 convolutions with residual connections [3] each followed by a strided convolution for downsampling. This results in a spatially low-dimensional representation of the image which is then passed to the decoder, which is a series of 5x5x5 convolutions with residual connections [3] and strided transpose-convolutions for upsampling. The input to each level of the decoder is an elementwise addition of the output from the previous decoder level and the cached encoder activations. This feature forwarding [11] from the encoder to the decoder is designed to supply fine-grained detail in reconstruction, and is an effective method for producing a higher quality network output. The shortcoming

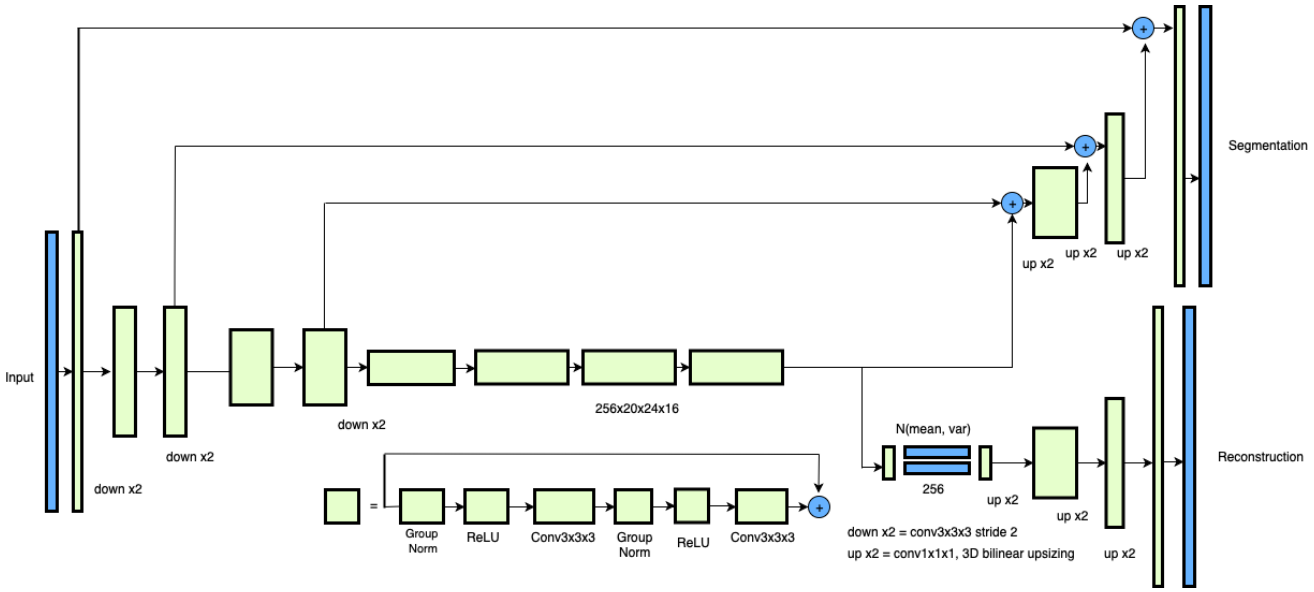


Figure 1: Convolutional encoder-decoder model architecture with VAE regularization

of the V-net is its use of larger convolutional filters instead of more layers.

2.2. U-net architecture

The U-net architecture [2, 11] presents a similar encoder-decoder model structure, but with a few important modifications. The U-net uses 3x3x3 convolutions with many more convolutional layers than the V-net does. This is more effective than the V-net since each filter is tasked with learning more local spatial interactions - stacking these allows for greater model expressivity. The efficacy of smaller convolutional filters can be supported by [12]. The U-net also elects to replace the strided convolution for downsampling in the encoder with a max-pooling layer. Empirically, these two downsampling methods have been shown to yield similar results. We opt to use max-pooling for downsampling since it reduces the number of trainable parameters without a significant tradeoff in performance, though [13] claims that convolutions are readily sufficient to capture downsampling features.

2.3. Adding a Variational Autoencoder

The model introduced in [10] is a U-net-like architecture with an additional variational autoencoder [7] branch. This model won the 2018 BraTS brain tumor segmentation challenge, attaining an average dice score of 82% across all 3 subtumor classifications [10, 1]. The addition of the VAE branch to the standard encoder-decoder architecture serves as an extra regularization constraint [10, 17] on the model, ensuring that the encoder is learning valuable feature representations in the latent space. Functionally, the encoder's downsampled output is fed to the VAE (as well as the decoder), which then uses this input to sample from a learned latent distribution, from which it reconstructs the original input image. During training, the reconstruction error is backpropagated to the layers of the encoder, which helps regularize it. Unfortunately, [10] does not provide comparative metrics to evaluate the impact of the VAE branch on out-of-sample performance, so we choose to adopt this scheme and can confirm its relative effectiveness on reducing overfit on the training set.

3. Methods

We construct our model based on the encoder-decoder with VAE architecture proposed by [10]. The model is trained on a weighted combination of multiple objectives (contributed by the decoder and the VAE). This architecture, unlike the U-net, has an asymmetrically deeper encoder to extract image features and a shallower decoder to reconstruct the segmentation mask. The model aims to produce a well reconstructed image in addition to the segmentation mask as a proxy objective to regularize the encoder of the network, and thus adds an additional branch with a variational autoencoder (VAE). We will first go over the base architecture from [10] then discuss changes we've made to it that have theoretical advantages.

3.1. Original Architecture

In [10], the authors apply image-wise normalization per channel to the input images. They run fully stochastic training with a batch size of 1 and optimize with Adam, schedul-

ing the learning per epoch:

$$\alpha = \alpha_0 \cdot \left(1 - \frac{e}{N_e}\right)^{0.9} \quad (1)$$

where α_0 is the initial learning rate $1e$-4, e is the current epoch number, and $N_e = 300$ is the total number of epochs to train for. The model is trained on a NVIDIA V100 32GB GPU. We replicate all of these aspects of training and preprocessing.

3.1.1 Encoder

The encoder uses an alternating series of ResNet [3] blocks and strided convolutions for downsampling. Each ResNet[3] block contains two convolutional layers followed by the additive identity skip connection. Each convolutional layer consists of [GroupNorm, Conv 3x3x3, ReLU] layers. Group normalization [16] is used instead of batch normalization due to batch size of 1 used to fit training in memory. The model downsamples the activation by a factor of two, three times in the encoder forward pass using strided convolutions. Each time the model downsamples, it also doubles the number of channels of the activation. The final encoder activation size is 20x24x16x256.

3.1.2 Decoder

The decoder performs the opposite dimensional operations as the encoder: spatial upsampling by a factor of 2 and halving the number of convolutional filters at each level. However, it dedicates half the number of ResNet blocks at each level. At the final output layer, there is a final pointwise convolution with a sigmoid activation to project the output to 3 channels, one for each possible kind of label. A dice loss is calculated for each channel, and the decoder loss is obtained by summing over the individual dice losses.

3.1.3 Variational Autoencoder

The VAE aims to reconstruct the original image, which will help ensure that the encoder is learning meaningful (and generalizeabe) representations, which is why it serves as a regularization branch. The VAE performs a strided convolution and a projection to a 256-dimensional space, the output of which is split into one representation each for the mean and variance from which an output is sampled. The VAE then projects this activation back to the original space and has the same convolutional structures passes as the decoder, but without encoder skip connections.

3.2. Architecture Modifications

We introduce several theoretical improvements to the architecture that we will discuss in this next section.

3.2.1 Halving Model Complexity

In our final model, we only use half of the number of convolutional filters used in [10]. This decision was made mainly due to additional memory constraints introduced by Tensorflow 2.0 (Tensorflow 1.x was used in [10]). We find that this reduction in model complexity did not provide any detriment to the model's ability to achieve baseline performance, however. We also believe that the difference in performance due to complexity is marginal, as the model already overfits to the training set with the downsized version.

3.2.2 Adding Squeeze-Excitation Layers

[5] introduces modifications to the original ResNet that shows improvements across the board. These squeeze-excitation layers are applied to the residual, which theoretically allows the model to learn the important portions of the residual to propagate through the block. This is intuitively similar to the highway networks introduced in [14], but are more tractable due to significantly less parameters.

The residual is first "squeezed" with a global average pooling. The squeezed residual is then projected onto a lower dimension with a reduced ratio of r followed by a ReLU, which "excites" the residual. This output is then projected back up to the channel size followed by a sigmoid. This is then used to scale the residual before its addition at the end of the block.

3.2.3 Initializing with He Normal

We also initialize our weights with He normal (except for those in sigmoid activation layers, which we initialize with Glorot normal), as this has been shown to lead to more stable training in deeper residual networks [4].

3.2.4 Adding Learning Rate Warmup

Because of our tripled dataset size, we found that adhering to the learning rate proposed in [10] led to divergence in training. We thus adopt a per-epoch linear warmup starting from $1e$-6 to $1e$-5 for the first 10 epochs. We warmup (and anneal) per epoch due to the noisiness of fully stochastic gradient descent

3.2.5 Downsampling with Max-Pooling

We also replace the strided convolutions with max-pooling instead. [10] downsamples with convolutions without mentioning any sort of normalization, so we witnessed an explosion in magnitude of the inputs through the deeper layers of the network. Rather than introduce normalization and activation layers, we opt for max pooling to reduce the number of parameters in the model. There is also no real consensus

on the difference in performance between the two downsampling methods in literature.

3.2.6 Reordering the Convolutional Layers

In [10], the convolutional layers were comprised of group normalization, then ReLU activation, and then a convolution. However, it is more common to structure these layers as convolutions followed by group normalization, then ReLU, as in[3]. This also allows us to keep the same cycle of convolution, normalization, and activation throughout the network, including single convolutional layers.

3.3. Hyperparameter Search

We primarily tuned the learning rate, which is crucial to convergence in training. We also experiment with the effects of increasing and decreasing the number of convolutional filters, as well as the number of ResNet (or SENet, perhaps) blocks that we use, much of which showed marginal changes in performance. We discuss our final hyperparameter decisions and rationale in greater detail in the Results section.

3.4. Objective Function

We formulate our loss identically as [10], as a weighted combination of three different loss terms:

$$L = L_{dice} + 0.1 \cdot L_{L2} + 0.1 \cdot L_{KL}, \tag{2}$$

where L_{dice}, L_{L2}, and L_{KL} correspond to dice loss, L2 reconstruction loss, and KL divergence loss, respectively. The 0.1 weighting on the VAE losses are derived empirically in [10].

3.4.1 Dice Loss

Because we want to optimize the dice coefficient, training the decoder on the corresponding objective function optimizes the score directly. Dice loss is an IoU (intersection over union) metric very similar to precision, as it tries to constrain the predicted set (of tumor voxels) to be as similar as possible to the ground truth:

$$L_{dice} = \frac{2 * \sum p \cdot \hat{p} + \epsilon}{\sum p^2 + \sum \hat{p}^2 + \epsilon}, \tag{3}$$

where p is the ground truth indicator, $\hat{p}$ is the predicted output probability, and $\epsilon = 1$ serves as some form of smoothing in the computation. We add dice loss functions over each of the three channels for each sub-region segmentation.

3.4.2 L2 Loss

L2 loss is the reconstruction loss contributed by the VAE branch which attempts to make the reconstructed image $\hat{R}$ as close to the original input image R as possible:

$$L_{L2} = ||\hat{R} - R||_2^2. \tag{4}$$

3.4.3 KL Loss

KL divergence is a way of measuring the difference between two probability distributions. We enforce this loss on the predicted mean and variance of the VAE to be as close to a standard Gaussian distribution as possible. The closed form loss function with respect to $\mathcal{N}(0, 1)$, the standard Gaussian, is as follows:

$$L_{KL} = \frac{1}{N} \sum \hat{\mu}^2 + \hat{\sigma}^2 - \log \hat{\sigma}^2 - 1 \tag{5}$$

where N is the total number of image voxels [10] and $\hat{\mu}$, $\hat{\sigma^2}$ are the mean and variance, respectively, of the learned distribution. Another important note is that in training, we learn $\log \sigma^2$ directly as opposed to σ^2, since the latter can yield negative variance values.

3.5. Metrics

While the primary metric to optimize is dice coefficient, we also track accuracy, precision, and recall in training, all of which provide intuitive explanations to what the model is learning. In this section, $\hat{p}$ is the prediction at a certain voxel and p is the corresponding voxel truth value.

3.5.1 Dice Coefficient

The dice coefficient is a common IoU metric used in semantic segmentation tasks. In our case, we report the average over the 3 separate dice coefficients calculated for each class label:

$$dice = \frac{2 \sum p \cdot \hat{p} + \epsilon}{\sum p^2 + \sum \hat{p}^2 + \epsilon}, \tag{6}$$

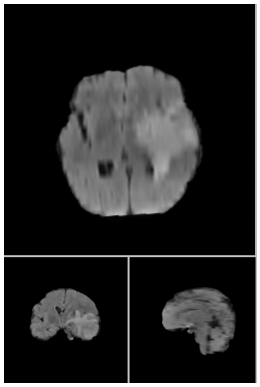
where $\epsilon = 1$ serves as some form of smoothing in the computation.

3.5.2 Accuracy

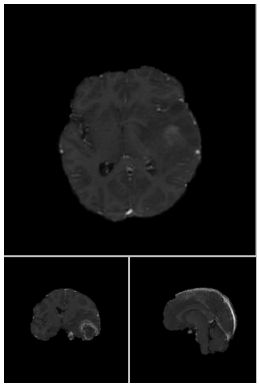
Accuracy is poor metric in our class, due to the severe class imbalance (the vast majority of voxels are tumor-free). Nonetheless, we track accuracy as a sanity check and expect the model to score highly on this metric. Accuracy is calculated as follows:

$$accuracy = \frac{\sum \mathbb{I}\{p = \hat{p}\}}{h \cdot w \cdot d} \tag{7}$$

where h, w, d are the spatial height, width, and depth, respectively.



(a) Data sample of axial (top), coronal (bottom-left), and sagittal (bottom-right) views of MRI scans of flair modality.



(b) Data sample of axial (top), coronal (bottom-left), and sagittal (bottom-right) views of MRI scans of t1-post modality.

3.5.3 Precision

Precision is a metric of evaluating the quality of the model's selections. It is computed as the rate of true positives with respect to all the positives predicted by the model. This is useful in this task for evaluating whether the model is overestimating or oversegmenting parts of the brain that do not contain tumors:

$$precision = \frac{\sum \mathbb{I}\{p \cdot \hat{p}\}}{\sum \hat{p} + \epsilon}. \tag{8}$$

3.5.4 Recall

Recall is a metric of evaluating the coverage of the model's selection. It is computed as the rate of true positives with respect to all the positives there are (in the true label). This is useful in this task for evaluating if the model is detecting all the tumords that it should be. A low recall score indicates that the model doing a poor job at predicting tumors when they are present:

$$recall = \frac{\sum \mathbb{I}\{p \cdot \hat{p}\}}{\sum p + \epsilon}, \tag{9}$$

4. Dataset

We use the dataset from the 2017 BraTS Challenge [1] which contains 285 labeled instances of 3-D MRI scans in 4 different modalities each [1]. Our training split is 260 examples and our validation split is 25 examples. Each instance contains a primary tumor lesion with additional necrotic (fluid-filled) sub-regions. Each spatial slice is a 2-D cross section of the brain from the top view with progressively deeper cross-sections along the depth dimension. All instances contain non-cancerous brain matter that the final model should learn not to classify as a tumor lesion. Unlabeled test set images are held by Stanford Medicine and will be human-evaluated by radiologists.

[1]Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby J, Freymann J, Farahani K, Davatzikos C. "Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-GBM collection", The Cancer Imaging Archive, 2017. DOI: 10.7937/K9/TCIA.2017.KLXWJJ1Q

Table 1: Validation metrics over 40 epochs

Epoch	Loss	Dice Score	Precision	Recall	Accuracy
0	2.438	0.049	0.576	0.905	0.057
10	1.459	0.401	0.634	0.990	0.502
20	1.217	0.560	0.668	0.993	0.639
30	1.195	0.587	0.666	0.993	0.660
40	1.124	0.620	0.666	0.994	0.691

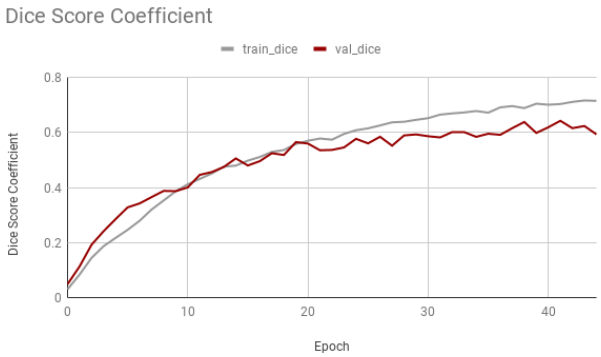


Figure 3: Training vs. validation dice score coefficient over 45 epochs

4.1. Preprocessing

We first perform a image-wise per-channel normalization excluding 0 valued pixels in computation of the mean and variance statistics. Though the paper performas a randomized intensity shift and scaling of image pixels across the training data, we believe that this introduces unnecessary complexity and noise to the task.

4.2. Augmentation

After data preprocessing, we randomly flip training instances with probability 0.75 across all spatiall axes. Next, we take three randomly placed spatial crops of size 144x144x128 on all data instances.

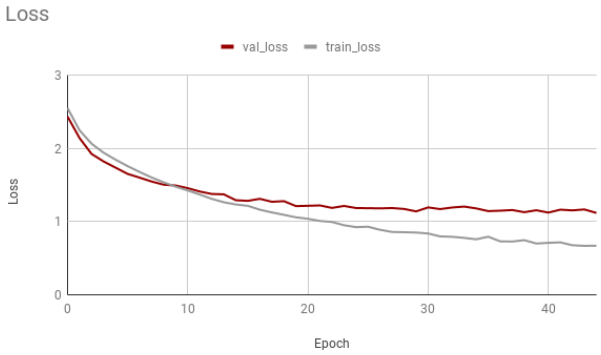


Figure 4: Training vs. validation loss over 45 epochs

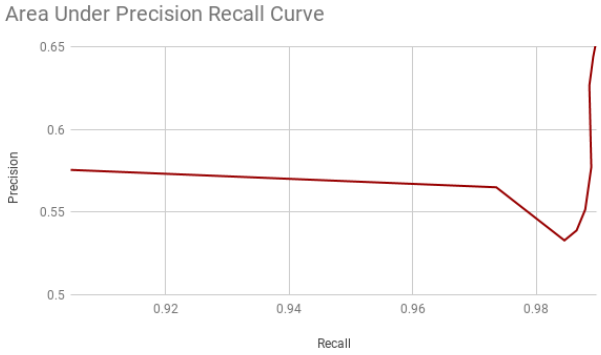


Figure 5: Area under precision recall curve, first 10 epochs.

5. Results

5.1. Hyperparameters

We made a number of hyperparameter choices in data preprocessing, on the model itself, and in optimization.

5.1.1 Preprocessing

In preprocessing we choose to apply image-wise normalization as opposed to pixel-wise normalization because pixel-wise normalization loses relative spatial information - the model struggled to reach 10% in early stages of training. We choose to mirror our inputs across each spatial axis with a probability of 0.75 as we saw that this provided sufficient data augmentation for the model to be robust on the validation set. Our final training set is comprised of 3 random crops per image to be used as training data (the same crops are made at validation), as opposed to 1 used in [10]. We decide to sample more crops to make our model and metrics less biased.

5.1.2 Model

For the model, we first choose to downsample with max pooling instead of strided convolution. Max pooling tends to outperform average pooling as is common for CNN architectures as explained in [15]. We opted away from strided convolution as done in [10] because it adds a large number of trainable parameters that do little to improve performance and make training more difficult. Correspondingly, we also choose to use linear upsampling for the decoder instead of strided deconvolution. We experiment with group normalization and batch normalization, finding that group normalization with a group size of 8 is preferable to batch normalization [16], which is ineffective with a batch size of 1. We use a filter size of 3 as is in [11, 10] as it facilitates more gradual information loss than a filter size of 5 [12]. A l2-regularization scale of $1e-5$ is sufficient to prevent severe overfitting to the training set.

5.1.3 Optimization

We train for a maximum of 300 epochs with early stopping and a linear annealing learning rate with a warmup for 10 epochs as described. We use a batch size of 1 as storing activations due to memory constraints.

5.2. Metrics

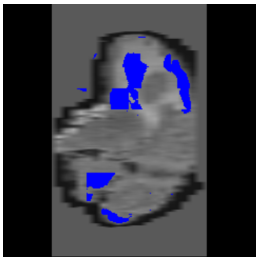
The primary metric we use is the dice coefficient, as it is the evaluation metric of choice in [1]. We also report precision and recall, which provide additional insights into the strengths as well as shortcomings of our segmentations.

5.3. Quantitative Results

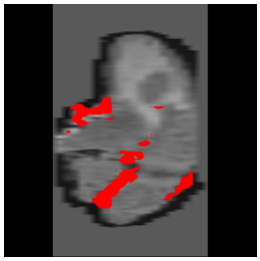
We first observe that we reach a dice score coefficient of 64% on the validation set after 45 epochs of training. Our training dice is slightly higher at 72% showing some overfit but an generally well-regularized model. This outperforms our baseline which achieved 60% dice score coefficient. We also observe a steadily declining loss curve over 45 epochs. As to be expected, our training loss is lower than our validation loss by about 0.5, however, the curves are sufficiently close to provide good out-of-sample performance. Our recall is nearly perfect, reflecting that the model is able to classify positive examples well. However, our precision is slightly lower, indicating a sizable number of false positives. However, compared to the baseline precision of 35%, we significantly improve the precision of the model, accomplishing the goals of the Stanford Cancer Institute.

5.4. Segmentations

First, comparing our output segmentations to the gold standard, we qualitatively observe that our masks do well

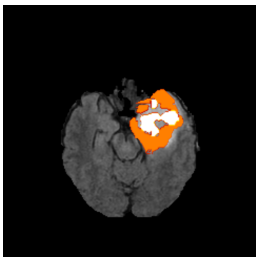


(a) Baseline segmentation on BraTS sample, flair modality, axial view (tumor core)

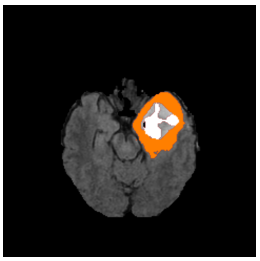


(b) Baseline segmentation on BraTS sample, flair modality, axial view (enchancing tumor)

(c) Baseline segmentation on BraTS sample, flair modality, axial view (whole tumor)



(a) Model segmentation on BraTS sample, flair modality, axial view



(b) Gold segmentation on BraTS sample, flair modality, axial view

(c) Model segmentation on BraTS sample, t1-post modality, sagittal view

(d) Gold segmentation on BraTS sample, t1-post modality, sagittal view

to approximate the true lesion region. Concretely, we accurately segment most of the orange, outermost, "whole tumor" subregion with undersegmenations in left side of the axial segmentation, as our lower precision metric suggests, and very minor oversegmentation on the right of the sagittal view. Our model also does well on the white, tumor core subregion with some minor breakages where there should be a wholly contiguous white subregion.

Compared to the existing baseline our model is vastly more precise. Each of the three subregion masks is displayed separately. Unfortunately, the baseline implementer was unable to provide us with ground truth segmentations for their data instances. However their model output segmentations demonstrate that, clearly, the tumor core segmentation is unable to highlight any particular well-formed region. Additionally, the whole tumor segmentation appears to struggle in differentiating between what should be labelled as whole tumor subregion and what should be background with no label.

6. Conclusion and Future Work

We confirm the effectiveness of encoder-decoder convolutional architectures in semantic segmentation as well as the regularization effect of the variational autoencoder on the encoder. We also conclude that learning rate warmup is essential when working with complex models and augmented data. Halving the number of convolutional filters throughout the model as compared to [10] evidently provided a simplification well-suited for the task. Using normalization techniques such has He initialization for ReLU layers as well as group normalization for activations facilitated in the convergence of the model. Adding Squeeze-Excitation layers also showed to be useful in offering additional information to activations deep in the network about which channel dimensions were most important to the computation. Lastly, reordering the convolutional layers and downsampling using max pooling as opposed to strided convolution did not present any clear disadvantages while simplifying the network and ensured that inputs were being normalized at each layer in the network.

There are many things that might be improved upon with our model. The focal loss, as proposed in [8], is designed to address extreme foreground-background class imbalance by preventing easy negatives, namely the large brain MRI background, from overwhelming the detector during training allowing it to focus on the sparse set of hard examples. We believe adding this objective to the modified dice loss formulation would help the model focus on oversegmented false positives and increase the model precision. Another avenue for experimentation is using feature visualization such as class activation maps as well as saliency maps to provide more interpretability to the model. This could aid in the creation of human-made features to supplement the hierarchical features learned by the network, as well as help professionals in the Stanford Cancer Institute better understand the model's decisions.

7. Acknowledgements

We would like to thank Haruka Itakura of the Stanford Cancer Institute for allowing us to work with her datasets and guiding us with a clear project scope.

References

[1] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki, M. Prastawa, E. Alberts, J. Lipková, J. B. Freymann, J. S. Kirby, M. Bilello, H. M. Fathallah-Shaykh, R. Wiest, J. Kirschke, B. Wiestler, R. R. Colen, A. Kotrotsou, P. LaMontagne, D. S. Marcus, M. Milchenko, A. Nazeri, M. Weber, A. Mahajan, U. Baid, D. Kwon, M. Agarwal, M. Alam, A. Albiol, A. Albiol, A. Varghese, T. A. Tuan, T. Arbel, A. Avery, P. B., S. Banerjee, T. Batchelder, K. N. Batmanghelich, E. Battistella, M. Bendszus, E. Benson, J. Bernal, G. Biros, M. Cabezas, S. Chandra, Y. Chang, and et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. *CoRR*, abs/1811.02629, 2018. 1, 2, 5, 6

[2] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *CoRR*, abs/1606.06650, 2016. 2

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 1, 3, 4

[4] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. 3

[5] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017. 3

[6] H. Itakura, A. S Achrol, L. Mitchell, J. J Loya, T. Liu, E. Westbroek, A. H Feroze, S. Rodriguez, S. Echegaray, T. Azad, K. W Yeom, S. Napel, D. Rubin, S. Chang, G. R Harsh, and O. Gevaert. Magnetic resonance image features identify glioblastoma phenotypic subtypes with distinct molecular pathway activities. *Science translational medicine*, 7:303ra138, 09 2015. 1

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014. 2

[8] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 7

[9] F. Milletari, N. Navab, and S. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *CoRR*, abs/1606.04797, 2016. 1

[10] A. Myronenko. 3d MRI brain tumor segmentation using autoencoder regularization. *CoRR*, abs/1810.11654, 2018. 1, 2, 3, 4, 6, 7

[11] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 1, 2, 6

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 6

[13] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014. 2

[14] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015. 3

[15] V. Suárez-Paniagua and I. Segura-Bedmar. Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction. *BMC Bioinformatics*, 19(8):209, Jun 2018. 6

[16] Y. Wu and K. He. Group normalization. *CoRR*, abs/1803.08494, 2018. 3, 6

[17] Y. Zhang, L. Li, and D. Wang. Vae-based regularization for deep speaker embedding. *CoRR*, abs/1904.03617, 2019. 2