DEEP LEARNING, CS395

DEEP BRAIN TUMOR SEGMENTATION WITH CLU-NET

May 7, 2018

Alexander Looi
alexander.looi@uvm.edu

Andy Metcalf
aametcal@uvm.edu

David Landay
dlanday@uvm.edu

Abstract

In this paper we reproduce results from Isensee et al. on the task of Glioma brain tumor segmentation using the 2017 Brain Tumor Segmentation (BraTS) challenge dataset. The Isensee 3D-UNet architecture has proved to be one of the most successful networks thus far for the BraTS challenge. Successfully replicating their work, we attempt to make improvements that address the excessive computational demand and lack of analysis relating to the correlation between individual MRI layers. We implement CLU-Net, a 2D-UNet on individual layers to generate the intra slice context followed by a convolutional LSTM (CLSTM) to utilize the inter-slice correlations. By downsizing to two dimensional convolutions, we are able to reduce the computational demand and thus train on larger batch sizes and build a deeper network. Initial results of CLU-Net seem promising, with general tumor segmentation yielding higher accuracy than the Isensee 3D U-Net model. These findings are suspect however, and further research must be done to validate and explain our initial results.

Introduction:

A quantitative assessment of brain tumors is essential for clinicians to make appropriate decisions regarding patient care. MRI scans form the basis of what doctors use to classify, identify, and assess the severity of different cancers in the brain. Performing such assessments on volumetric MRI scans is time consuming and resource intensive because a skilled health care worker must look through and manually label entire volumes of 2-D layers. Even though adjacent layers may appear identical, accurate maps and segmentation of the brain tumor and its structures are very important because these segmentation maps are used in targeted treatments such as chemo, and radiation therapy. In addition to skilled labor time, a quantitative assessment of brain tumors, gliomas in particular, is quite difficult due to the heterogeneity of the tumors appearance and shape. Thus, there can be considerable disagreement among humans trying to classify such tumors (74 - 85% agreement rate for our dataset). Therefore, software that can consistently, quickly, and accurately, identify and segment structures in glioma tumors, can be a great aid to oncological health care professionals. The Brain Tumor Segmentation Challenge (BRATS [4]) was initiated to address these challenges by employing computer algorithms to automatically segment these MRI scans.

Addressing this segmentation challenge, we first employ a three dimensional analog of the U-Net architecture first proposed by Ronneberger et al. [5]. This architecture was chosen for it's high accuracy in both the 2015 [5] and 2017 [3] BraTS challenges. Although accurate, the large number of trainable weights make this architecture both slow to train and memory intensive. To address this memory demand batch size must be reduced, which likely limits the learning capacity of the network.

In addition to the 3D-UNet, we implement the original 2D-UNet coupled with a recurrent neural network (RNN) to boost the performance of the network by alleviating some of the major problems (including memory demand) associated with the 3D-UNet. In this architecture the 2D-UNet will gather information relating to the individual brain slices within the 3D volume and the RNN will extract information relating individual concurrent slices within a MRI volume.

Below we discuss previous work linking convolutional neural networks with recurrent neural networks, we then follow with our methodology, description of our dataset, and results. Finally, we end with a discussion of our results and future research.

Literature Review:

To perform a similar task in biomedical MRI image segmentation Chen et al. [1] combine a Fully Convolutional Neural Network (FCN) [6] with a RNN, using the FCN to extract the intra-slice context and the RNN to extract the inter-slice context. Generally, the authors of this paper utilize a similar structure to our 2D UNet / RNN network architecture. MRI slices are fed into a UNet type structure, followed by a RNN that outputs to a segmentation probability map. Taking a slightly more complex tack, the authors of this paper replace the traditional UNet with a 'k-UNet' consisting of multiple UNets linked together, many having downsized inputs to address varying scales of image content within a 3D MRI volume. Additionally they replace the RNN with a Bi-directional Convolutional LSTM. The bi-directional aspect takes into account context from both z-directions of the volume, and the Convolutional LSTM is an analog to the traditional LSTM, with convolutions taking the place of vector multiplication at the LSTM gates. To validate their network, the authors also implemented other algorithms such as Pyrimid LSTM, U-Net and 3D convolution to compare to their RCN-CLSTM network. Not surprisingly, the FCN-CLSTM achieved the highest accuracy, however the addition of the CLSTM only slightly improved the accuracy over the FCN by itself.

Visin et.al [8] also combine a Fully Convolutional Network and Recurrent Neural Network in their segmentation scheme. Here, they extend the ReNet model [7], originally designed for image classification tasks, to perform semantic segmentation. The ReNet layer replaces the convolutional+pooling layers of a traditional CNN architecture with a series of 4 RNNs that "sweep" vertically and then horizontally over an input image. In so doing, the ReNet layer has significant advantages over it's predecessors (see citations section of:[8], [7]) as it utilizes a uni-dimensional RNN as opposed to resource intensive multidimensional architectures (see citations section of: [8], [7]), and preserves local contexts at the pixel level, simultaneously providing global features.

The ReNet layer takes an input image $X = \{x_{ij}\}$: $X \in \mathbb{R}^{w \times h \times c}$, where w, h, and c denote the width, height and channel dimensions of the input or feature map respectively, and looks at the recurrent relationships among all pixels in a single direction by first sweeping vertically across specified non-overlapping patches derived from the input. The output hidden states are then concatenated together and a similar horizontal sweep is performed over this composite feature map. For classification tasks, the output of the final step is flattened and fed to a softmax classifier. The authors of ReSeg feed the output of a VGG-16 model, pretrained on the ImageNet dataset, into a series of 4 ReNet layers stacked on top of one another. Specifically, the authors chose the gated recurrent unit (GRU) as the activation layers

in the network as an added resource conserving element. It should be noted that the ReNet layers are easily parallelizable as each RNN is dependent upon pixels in a particular direction.

To perform image segmentation, the feature maps are upsampled using transposed convolutions, and then the IoU score is computed between the predicted segmentation mask (given by a softmax layer) and a ground truth mask. While this methodology is a novel example of utilizing a FCN-RNN architecture, it differs greatly from the task we are attempting to complete. For one, we train our model on a separate accuracy metric which accounts for the class imbalances inherent to our problem. In addition, the goal of the RNN employed in our network is to learn both temporal and spatial dependencies among the different layers of a 3D brain scan, which is different than the pixelwise spatial dependencies a ReNet layer attempts to learn over the input image itself.

Methodology:

Data:

We trained our model using the mutlimodal Brain Tumor Image Segmentation Benchmark (BraTS) dataset, which is comprised of 3D volumetric brain scan images. When this data set was first introduced, it included only 65 multi-contrast Magnetic Resonance (MR) scans of both high grade (HGG) and low grade (LGG) glioma patients [4]. Since then, the data set has been expanded to 210 HGG and 75 LGG patients. For each patient there are four different image types: T1, T1c, T2, and FLAIR. Each, type of MRI image enhances, via black and white contrast, different tissues of the human body. In T1 images human tissue such as fat, white matter (a type of brain tissue), or bone marrow will appear bright; while air, bone, and fluids will appear dark. In T2 imaging, fat and bone marrow will still appear bright however, fluids will appear as bright (in contrast to T1 imaging) while other tissues may appear darker. Finally, in FLAIR imaging the fluid signal is nullified, which can help clean up tissue images that are typically surrounded by fluid. Thus, each imaging type will accentuate different aspects of the human body and will provide a CNN with new and interesting information each time [4].

Tumors are present in every brain scan in this dataset, however, each image has a variable tumor size and placement. Additionally, images are high enough quality that the four structures of the glioma tumors can be identified. All tumors are “expertly annotated”,

meaning that trained clinicians rated, identified, and marked the tumors and the specific parts of the four glioma structures based on a set of standard criteria. Different parts of the tumor structures can be identified by studying the different MRI modality types provided in the data set. Here the goal is to accurately segment these brain tumors into the appropriate segmented categories and the correct segment boundaries.

Instead of looking at each MRI scan type individually, we utilize the same approach taken in the Isensee et al. paper [3] by treating each MRI scan type as a separate channel; stacking the four corresponding layers for each scan type together. Thus, our data set for each brain consists of a tensor shape of 155 layers, with 240 × 240 images, each with 4 channels. This way the CNN portion of the recurrent neural net, will look through all four channels and build corresponding constitutional filters. There is good information in all four MRI scan types that need to be considered because human glioma classifiers will consider all four MRI scans at once. Our formulation of the data also reduces any initial biases during the first several epochs of training when a specific modality type is trained on first. Lastly, we conduct batch normalization at training by normalizing values from 0 to 1.

Proposed Architecture:

For our initial model we used the 2D U-Net model first proposed by Ronneberger et al. [5], in 2015 and then modified by Çiçek et al. in 2016 [9]. The U-Net proposed by Ronneberger et al. is a 2D CNN encoder decoder architecture where all convolution layers are 3×3 feature extractors. Subsequent feature maps are taken down to smaller resolutions through 2×2 max pooling layers, two convolutional layers were used between each max pool layer with varying number of filters. This is the encoding portion of the CNN. Once the feature map reaches a 6×6 resolution (if inputs start off at 96 × 96), the images are "up-conv-ed" (or upsampled) through a 2×2 up-convolution, after upsampling two convolutional layers were used.

Here, the upsampling portion of this CNN is the decoder. In addition, to the upsampling, before each max pooling layer all generated feature maps are copied and "attached" (concatenated) to the corresponding feature maps just after upsampling is performed and before the normal 3×3 convolutions. These attached feature maps provide a "memory" of previous feature maps to aid in the CNN decoding. This model was the best model for the 2015 ISBI cell segmentation tracking challenge and can be trained end to end with few images.

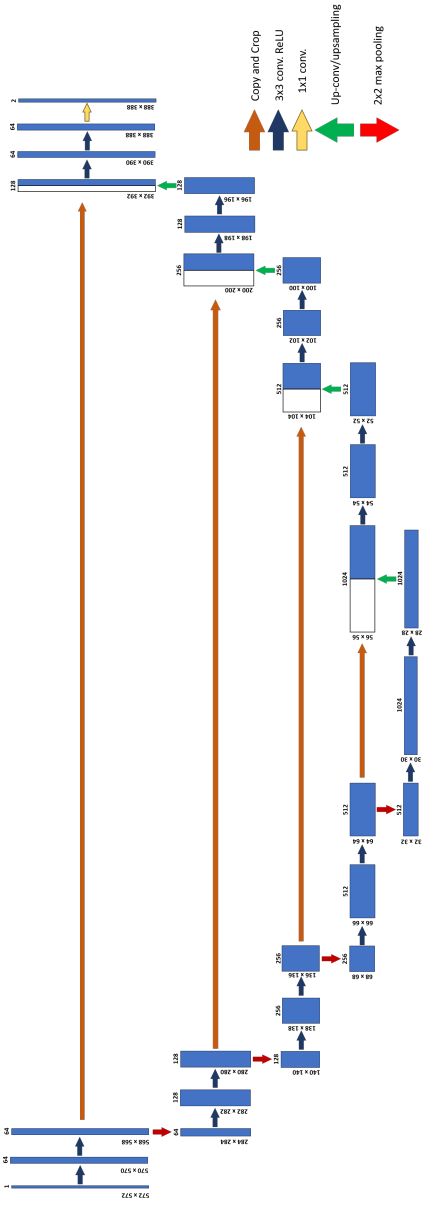


Figure 1: The original 2D U-Net architecture. Blue boxes are multi-channel feature maps. Arrows so the direction of convolution, upsamples, and copy pastes. Numbers indicate the the feature-map dimensions.

A year later in 2016 Çiçek et al. published a 3D U-Net model where all 2D 3×3 convolution layers were replaced with 3D 3×3×3 convolution layers and all 2D 2×2 max pooling layers were replaced with 2×2×2 max pooling layers. In 2017, Isensee et al. produced a U-Net inspired 3D CNN to specifically address the 2017 BraTS dataset and brain tumor segmentation challenge. Here they kept the CNN encoder decoder architecture with concatenation of encoder feature maps to the end of decoder features but added segmentation layers to the decoder portion of the CNN (a form of "deep supervision"). These segmentation layers provide additional context to the final segmentation where the loss associated with these added segmentations are weighted and added to the final loss function. With this final architecture Isensee et al. were able achieve accuracies of 89.6, 79.7, and 73.2% on the different classes of tumor structure segmentation. Using their provided code, data augmentation procedures, and dataset we reimplemented their model and trained it from scratch to see if we could achieve similar accuracy rates.

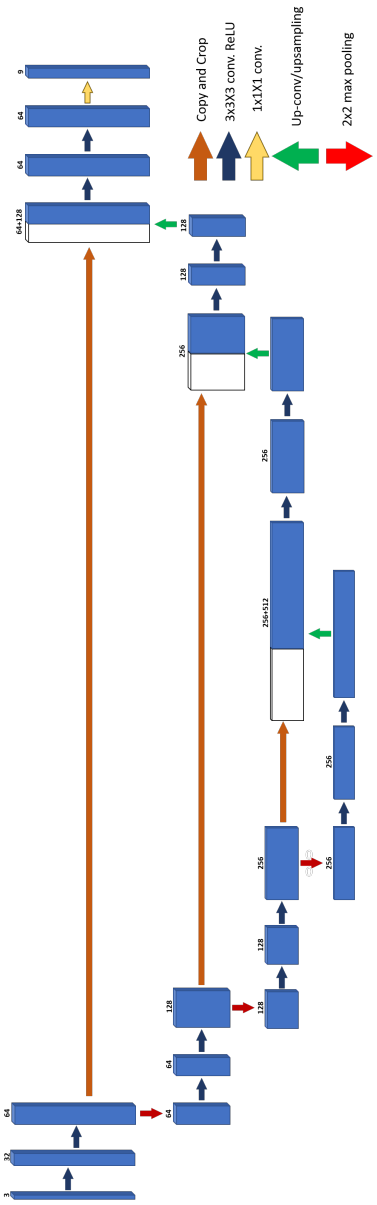


Figure 2: The modified 3D U-Net architecture. Blue boxes are multi-channel feature maps. Arrows so the direction of convolution, upsamples, and copy pastes. Numbers indicate the the feature-map dimensions. Note that the 3D UNet model has fewer layers.

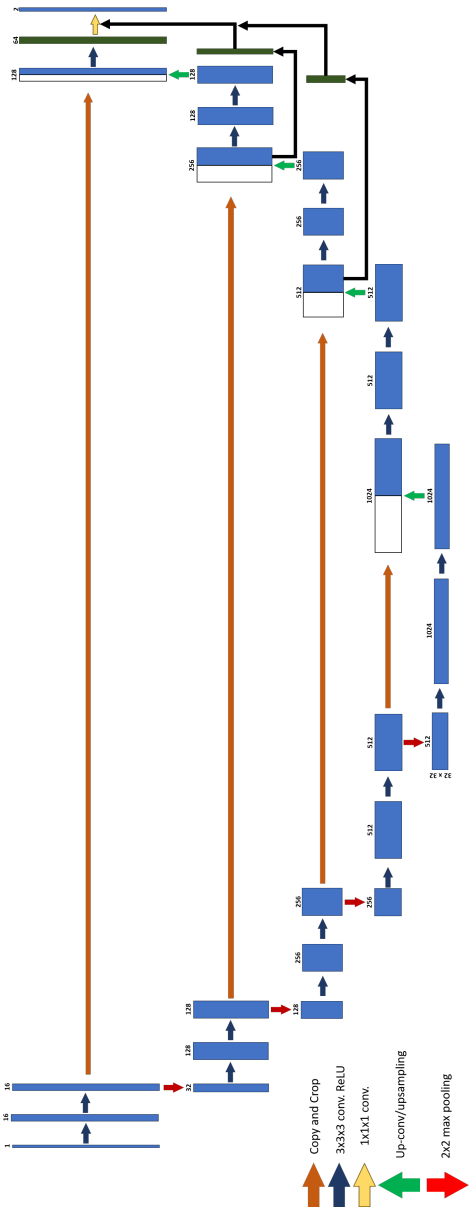


Figure 3: The more modified 3D U-Net architecture proposed by Isensee. Blue boxes are multi-channel feature maps. Arrows so the direction of convolution, upsamples, and copy pastes. Green boxes are segmentation layers, which provide additional context for the CNN and the decoding layers. Numbers indicate the the feature-map dimensions. Note that the 3D UNet model has fewer layers.

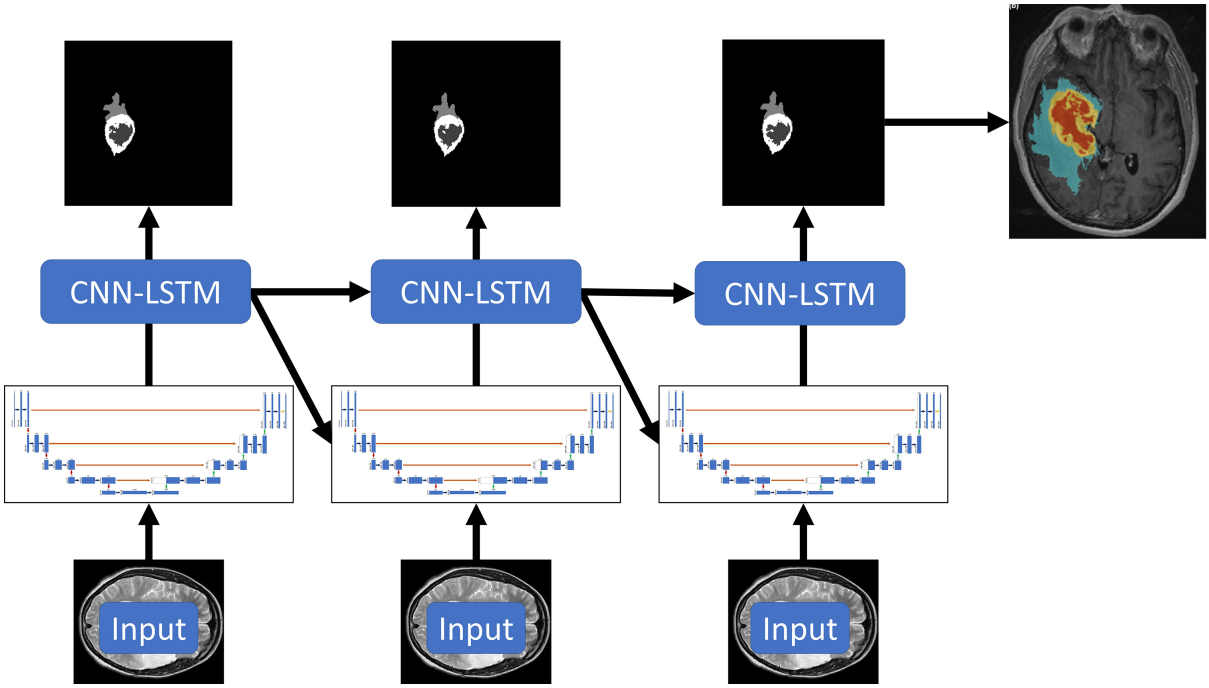


Figure 4: The CLU-NET architecture, which conducts semantic segmentation on 2D layers that have multiple layers. This uses a U-Net as a feature extractor/encoder/decoder, then a CNN-LSTM to do segmentation, and pass that information to the next iteration. This model is indented to

Although the Isensee model is considered state-of-the-art and has achieved excellent results in recent BraTS challenges, it does not come without flaw. The 3D convolutional filters contain a vast amount of weights for the network to learn, resulting in slow training and large memory consumption. Because of this, batch size is reduced considerably, which likely decreases the learning capability of the network. Additionally, simply linking the 2D UNet with its 3D analog does not leverage the spatial correlation along the z-axis of the MRI volumes. Clearly these volumes are sequential in nature, so the correlation between slices should provide valuable information for segmentation.

To address these issues, we implement a 2D UNet with a Convolutional LSTM (CLSTM) built on top. This provides additional information from previous layers to the segmentation of future layers. As mentioned above, individual, sequential slices are fed into the 2D-UNet which extracts the intra-slice context and the resulting features are then passed into the CLSTM to extract the inter-slice context between layers. The 2D-UNet consists of four encoder convolutional layers followed by four decoder up-convolution layers, all having zero padding to maintain dimensionality for lateral concatenations and 3x3 filters.

The CLSTM is placed on top with a softmax classifier to produce the segmentation. We chose this architecture specifically because it addresses the two major pitfalls associated with the 3D-UNet. Memory consumption and training time should be dramatically reduced by using 2D convolutions in place of their 3D counterparts, and intra-slice context that is likely overlooked by the 3D-UNet should be taken into account with the CLSTM layer. As a control, to see how the CLSTM layer contributed to the above architecture, a 2D-UNet without the CLSTM was also implemented. We call this U-Net CNN-LSTM architecture CLU-Net.

Model Training:

CLU-Net was trained for 400 epochs, with a batch size of 20, using a step size of 10 (this refers to the number of images that are included per batch). Images were down-sampled from 240 × 240 to 96 × 96 pixel images. Batches of images were normalized on the fly during training. Our learning rate was 1e-7.

Loss Function:

To measure model performance we used a modified version of the Dice Loss function, a metric that measures the overlap (similarity) between two sets and is similar to that proposed by [2]. It is a metric commonly applied in medical image segmentation tasks where there exists class imbalances among different regions of the input image, and a ground truth mask is present. In our network, a softmax layer outputs a probability for each pixel in the image belonging to a particular class. In our case, 5 class labels are used to distinguish between background and specific tumor structures, thus predicting a mask of the tumor region. The predicted mask is then compared to a corresponding ground truth by measuring the overlap between them.

Traditionally, the Dice Loss function for a binary classification task is denoted:

$$\mathscr{L}_{dc} = \frac{2\sum_i p_{ij} y_{ij}}{\sum_i p_{ij} + y_{ij}} \tag{1}$$

where p_{ij} represents the predicted softmax probabilities over a pixel i and y_{ij} denotes the corresponding ground truth value of pixel i for an instance j.

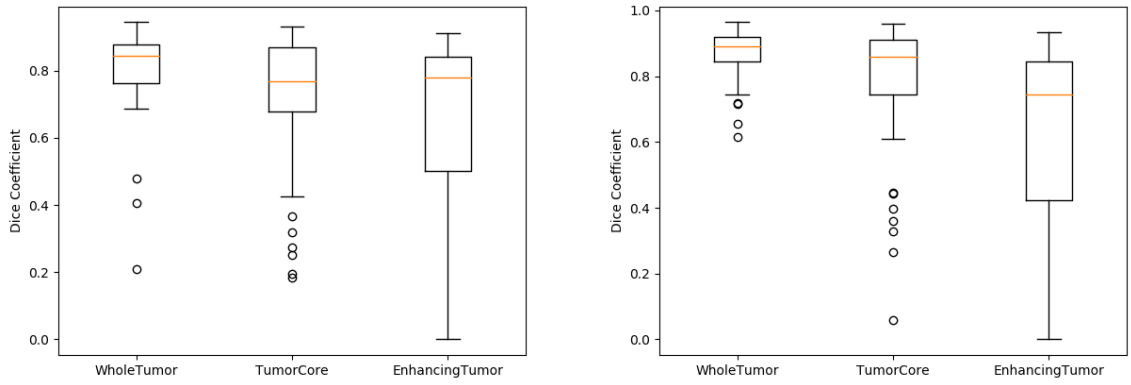
The Isensee paper defines a multiclass, differentiable, Dice loss function by:

$$\mathscr{L}_{dc} = -\frac{2}{|K|}\sum_{k\in K}\frac{\sum_i u_i^k v_i^k}{\sum_i u_i^k + \sum_i v_i^k} \tag{2}$$

“where u is the softmax output of the network and v is the one-hot encoding of the ground truth segmentation map" see [3]. Here, $k \in K$ are the different classes. We use this same variation of the Dice loss function to determine the quality of our own segmentation, with the exception that we substitute the set of voxels in a training volume, $i \in I$, for the set of pixels in a training image.

Results:

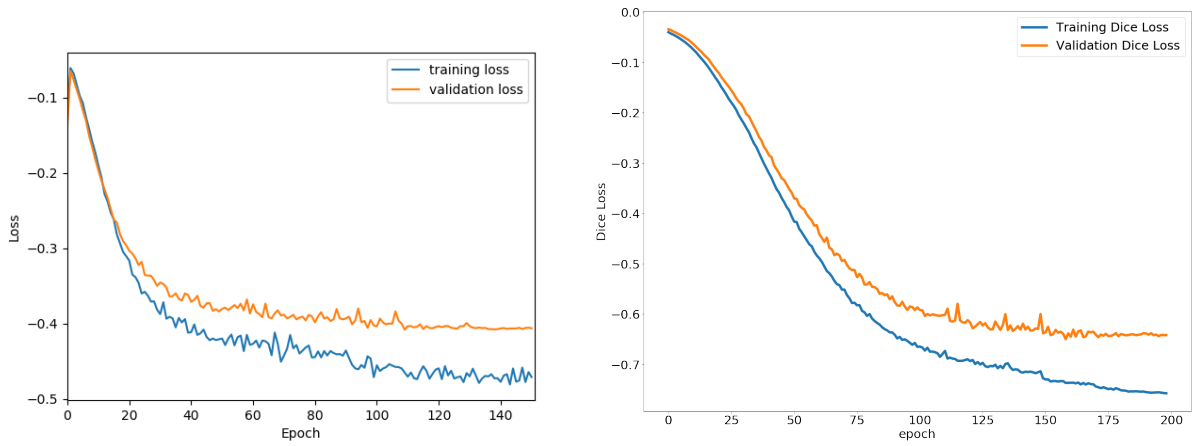
3D U-Net



(a) Our Validation Scores (64,64) patch size

(b) Full image validation scores

Figure 5: Dice score comparison of first and second trial of 3D-UNet



(a) Dice loss on (64,64) patch size

(b) Full image dice loss

Figure 6: Dice loss comparison of first and second trial of 3D-UNet

We implemented the Isensee 3D-UNet model two times, first with a reduced patch size of 64 x 64 and batch size of one for sorter training time, and second on the entire image, still however with a batch size of one brain. In the final trial where we trained on full images, an initial learning rate of 5e-4 was used and we trained for a total of 200 epochs. The results are shown above in figures 5 and 6. When training on the full image with no patch we get similar and perhaps slightly better validation accuracy than the original Isensee paper which had dice accuracy scores of 0.896, 0.797, and 0.732 for whole tumor, tumor core, and enhancing tumor respectively. We find a similar pattern with the whole tumor and tumor core being easier to predict than the enhancing tumor. This is likely because the enhancing tumor is the dynamic, changing portion of the tumor that requires the FLAIR contrast to be identified by humans. As we can see from the boxplots, the enhancing tumor segmentation has the highest dice score variability, indicating that the region is difficult to predict. It is also clear from our loss plots that the validation loss converges fairly well by epoch 200. We did not train for additional epochs or increase the batch size due to the computational expense. At approximately 27 minutes per epoch, the 3D-UNet took nearly four days to train on a Nvidia Tesla k80.

2D U-Net

The Dice loss and accuracy for the 2D U-Net did not converge after 400 epochs with step sizes of 42, with each step having a batch size of 155. Accuracy and Dice loss values

started low and remained low (less than 0.001).

Recurrent U-Net: CLU-Net

Overall, the Dice loss for the CLU-Net converged to about -0.9 for both training and validation. Looking at figure 3 we see that the training loss varies quite wildly particularly in comparison to the validation loss. Additionally, this is validation loss for the entire tumor and does not provide any information on how well our model does for individual tumor structures. Thus, for general tumor classification we do better than the Isensee 3D U-Net model.

Additionally, our model trains much faster on both a per time step and a general basis. On an Nvidia Tesla k80 each epoch took about 72-75 seconds for a batch size of 20 with 42 steps. So training for 400 epochs only takes 8 hours compared to the 62 hours the Isensee model took.

Lastly, because we were not officially part of the Brats Competition thus we do not have the truth masks for the test data and are not presently able to do fully conclusive tests to compare class-wise segmentation to the Isensee 3D U-Net.

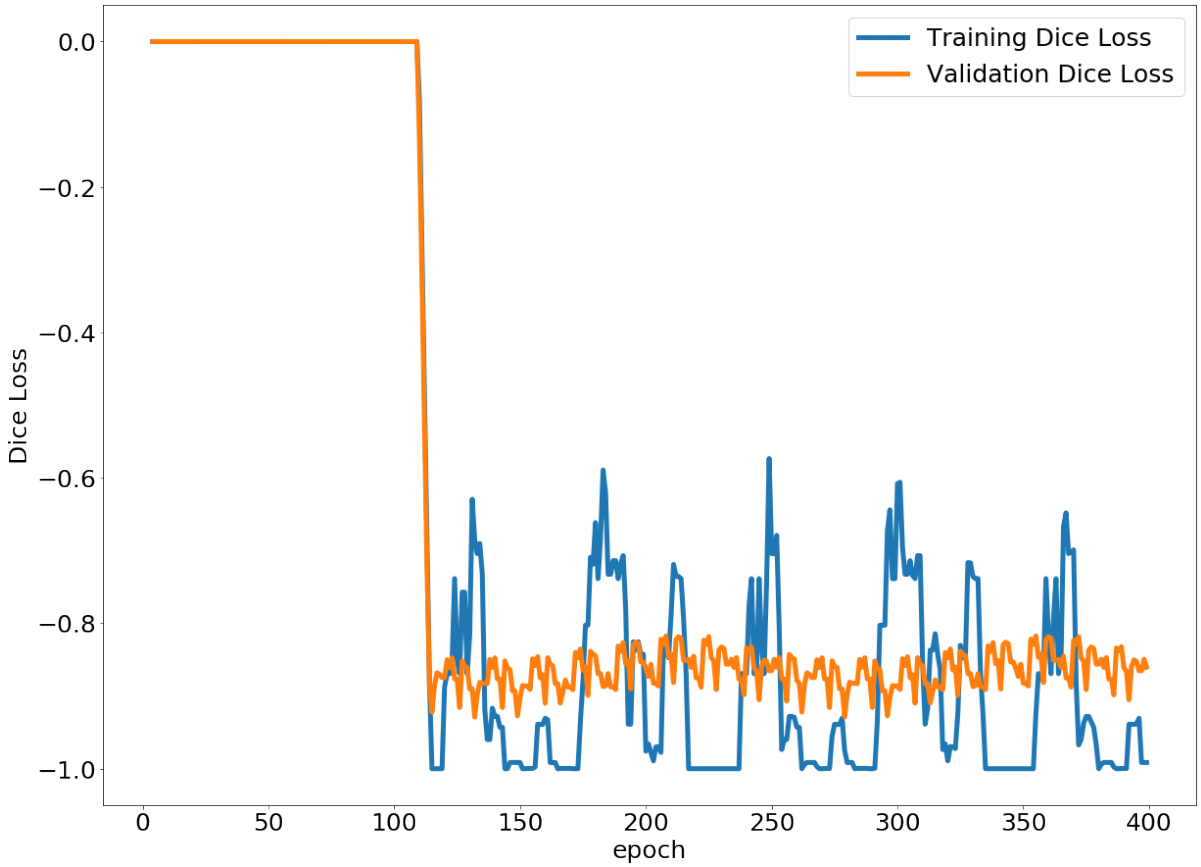


Figure 7: The Dice loss for our Convolutional LSTM U-Net Model. Learning rate was very low (α = 1e-7). The model converges unusually quickly, with the training loss fluctuating quite wildly.

Table 1: Table comparing the number of parameters, the training time per epoch, batch sizes, and validation loss of the Isensee 2017 model and CLU-Net. The validation loss, number of parameters to train, and the time per epoch is much better for CLU-Net. Additionally, CLU-Net can be trained with a larger batch size.

Model	Number of Parameters	Time per Epoch	Batch Size	Validation loss
Isensee 2017	Not Reported	27:00	2	0.69
CLU-Net	33 million	1:12	40	0.91

Discussion and Conclusions:

By using a recurrent 2D CNN architecture, we are able to make the convolutional portion of our model deeper compared to other architectures. In the Isensee paper they have seven total layers (3 encoding layers and 3 decoding layers). With CLU-Net we are able

to go deeper with 11 total layers (5 encoding layers, 5 decoding layers). We could tune the training time and model quality by adjusting the number of step sizes the model trained on. Increasing the step size (number of recurrent images to train on) increases the training time. The results presented in the paper here, we used a step size of 10 images.

CLU-Net is not perfect. Additionally, our loss curves are suspicious. The loss suddenly converges in only a couple of epochs which suggests that it may have found a local optima. Because of the sudden convergence we trained CLU-Net an additional time to see if we could replicate the results. Upon a second training we achieved similar results with a similar looking loss. Secondly, the training loss varies wildly compared to the validation. This may indicate that we are over fitting the model, thus some data augmentation may be necessary to artificially provide additional data. Lastly, we may need to use a larger step size or batch size. The seemingly periodic spikes in training loss and validation loss indicate that these hyper-parameters are too small.

From the Isensee paper, we see a decent amount of separation between the validation loss and the training loss suggesting minor over fitting, however, their test results do indicate that this may not be the case. Without submitting our own predictions to the Brats competition we are unable to say whether our model performs better than the Isensee one.

Our model does out perform the Isensee paper definitively in a number of ways. Our model trains faster, probably has less parameters (Isensee did not report their parameter count), and uses less memory during training and on prediction.

It is important to note that training speed nor memory usage on training and inference are not necessarily important parameters in a cancer segmentation problem. The most important metric is often accuracy because typical treatments for cancer can be highly targeted (i.e. radiation treatment or surgery). In these cases (particularly if you cannot open up a person to view the tumor) accuracy is very important because we want to avoid irradiating healthy tissue.

If a model takes several minutes versus a couple of seconds on inference this is not necessarily a bad trait either because a Doctor viewing and segmenting tumors can take equally as long. These models in their current state can be important pre-screening tools that aid doctors in the diagnosis of tumors and should not in their current form be the sole mechanism for diagnosis. They can also be used to estimate tumor size/volume and generally what tumor structures are present.

That said, training time may become important if machine learning algorithms are to be used in real time. For example, surgical tasks could make use of real time segmentation to assist and inform physicians during various procedures. Current accuracies and training times are not there yet, but future progress may yield more applications.

Future Work:

All of the models presented here are supervised problems. However, these models have the potential capacity to learn of structures or features that doctors have not noticed yet (provided a model is deep enough) Thus, we can try and use unsupervised methods to learn new features, and use visualization techniques to see what these features may be. In addition, a lingering issue of semantic segmentation for medical images is that an extensive decision making process must be employed to produce accurate ground truth images. It is possible that an unsupervised solution could contribute to the consensus of gold standard images.

Because of the small amount of data provided in the original Brats data set, we would like to do some data augmentation. Some of the evidence of over fitting we see could be a result of lack of data. We were not able to implement data augmentation currently because of the unique shape of our data set. We would have to write our own data augmenter to implement data augmentation at training, given more time we would have done this.
Lastly, we want to give our model brains that have either other diseases or are disease free to test its ability to distinguish tumors from other artifacts. It could be that tumor structures share characteristics with other brain diseases, so it would be interesting to see how our model performed on entirely different brain diseases.

Bibliography

[1] Jianxu Chen, Lin Yang, Yizhe Zhang, Mark Alber, and Danny Z. Chen. Combining Fully Convolutional and Recurrent Neural Networks for 3d Biomedical Image Segmentation. In *30th Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016.

[2] W. R. Crum, O. Camara, and D. L. G. Hill. Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461, November 2006.

[3] Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein. Brain Tumor Segmentation and Radiomics Survival Prediction: Contribution to the BRATS 2017 Challenge. page 8, 2017.

[4] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elizabeth Gerstner, Marc-André Weber, Tal Arbel, Brian B. Avants, Nicholas Ayache, Patricia Buendia, D. Louis Collins, Nicolas Cordier, Jason J. Corso, Antonio Criminisi, Tilak Das, Hervé Delingette, Çağatay Demiralp, Christopher R. Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andac Hamamci, Khan M. Iftekharuddin, Raj Jena, Nigel M. John, Ender Konukoglu, Danial Lashkari, José António Mariz, Raphael Meier, Sérgio Pereira, Doina Precup, Stephen J. Price, Tammy Riklin Raviv, Syed M. S. Reza, Michael Ryan, Duygu Sarikaya, Lawrence Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos A. Silva, Nuno Sousa, Nagesh K. Subbanna, Gabor Szekely, Thomas J. Taylor, Owen M. Thomas, Nicholas J. Tustison, Gozde Unal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koen Van Leemput. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Trans Med Imaging*, 34(10):1993–2024, October 2015.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351, pages 234–241. Springer International Publishing, Cham, 2015.

[6] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, April 2017.

[7] Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, and Yoshua Bengio. ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks. *arXiv:1505.00393 [cs]*, May 2015. arXiv: 1505.00393.

[8] Francesco Visin, Adiana Romero, Cho Kyungyun, Matteo Matteucci, Marco Ciccone, Kyle Kastner, Yoshua Bengio, and Aaron Courville. ReSeg: A Recurrent Neural Network-based Modelfor Semantic Segmentation. *IEEE*, 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW):426–433, June 2016.

[9] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, volume 9901, pages 424–432. Springer International Publishing, Cham, 2016.