Multimodal Brain MRI Tumor Segmentation via Convolutional Neural Networks

Liyue Shen
Stanford University
Stanford, CA
liyues@stanford.edu

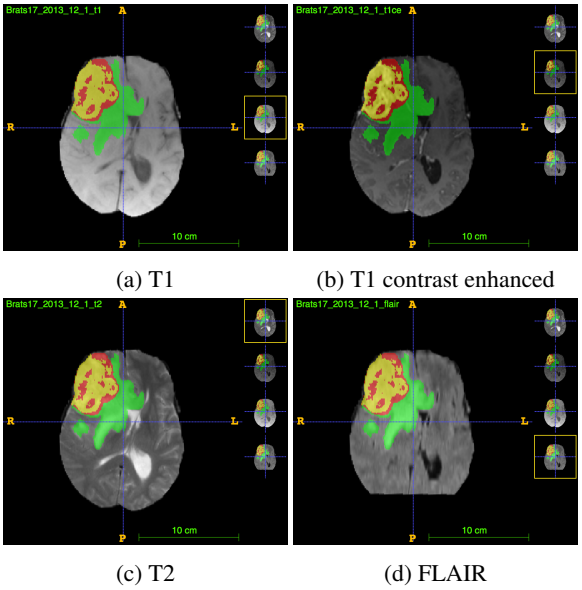
Timothy Anderson
Stanford University
Stanford, CA
timmya@stanford.edu

Abstract

Glioma are the most common family of brain tumors, with a subset of glioma known as glioblastoma forming the most common and some of the highest-mortality and economically costly forms of brain cancer. Patients are diagnosed based on manual segmentation and analysis of multimodal MRI scans, but due to the labor-intensive nature of the manual segmentation process and mistakes or disagreement between manual segmentations, there exists a need for a fast and robust automated segmentation algorithm. Convolutional neural networks (CNNs) have been shown to be extremely effective for a variety of visual recognition and semantic segmentation tasks. Here, we present three novel CNN-based architectures for glioma segmentation for images from the MICCAI BraTS Challenge dataset. We also explore transfer learning between the BraTS dataset and other neuroimaging datasets by applying models pre-trained on the BraTS dataset to segmenting images from the Rembrandt dataset. Our results show that patch-wise approaches trained on a balanced training set of tumor and non-tumor patches delivers strong segmentation results with mean dice score of 0.86. The results from transfer learning show that applying models pre-trained on the BraTS dataset to other neuroimaging datasets is promising but requires further work.

1. Introduction

Brain tumors have an average incidence rate of 26.55 per 100,000 for women and 22.37 per 100,000 for men [12]. Gliomas are the most commonly occurring type of brain tumors and are potentially very dangerous [15, 9], with about 90% of gliomas belonging to a highly aggressive class of cancerous tumors known as glioblastomas [30]. Glioblastoma is the most common form of brain cancer and is highly aggressive, with a 5 year survival rate of 5.3 % for patients aged 40 to 64 [12] and median survival time of 331 to 529 days (depending on the course of treatment) [20]. In addition to high mortality rates, glioblastoma is very costly to treat, with a mean expenditure of over $100,000 in the six months post surgery [20]. Consequently, there exists a significant need to accurately diagnose gliomas and glioblastomas in their early stages.



(a) T1 (b) T1 contrast enhanced

(c) T2 (d) FLAIR

Figure 1: BraTS dataset images. Images from ITK-Snap.

Multimodality magnetic resonance imaging is the primary method of screening and diagnosis for gliomas [30]. Accurate segmentation of the tumor to determine features such as volume, spread, and location is critical to diagnosis and forming a course of treatment [15, 10, 31]. Currently, tumor regions are segmented manually by radiologists, but advances in computer vision have made possible the ability to automate the segmentation process. Specifically, tumor segmentation algorithms based on convolutional neural networks (CNNs) have been shown to be at least as effective as other automated tumor segmentation methods [10].

Here, we present a novel approach to glioma segmentation based on deep neural networks. We present two patch-wise CNN architectures for patch-wise binary classification

of tumor and non-tumor regions, and a full-image CNN architecture. We will train and test both architectures on the BraTS Challenge dataset [15] (example images shown in Fig. 1), and explore transfer learning to the Rembrandt dataset [24]. Due to the relatively small size of the data sets involved, we also explore several methods to prevent model overfitting and improve robustness.

In the following, we introduce a brief overview of previous work for biomedical image segmentation and transfer learning. We then propose and evaluate our model architectures for tumor segmentation. Finally, we present results for transfer learning between neuroimaging datasets.

2. Related Work

Beginning with the success of [13], in recent years there has been a growing interest in using CNNs for image classification and segmentation. Motivated by the efficacy of CNNs and other deep architectures across computer vision tasks, there is much interest within the medical community in applying CNNs to medical image processing problems due to the high accuracy and throughput possible with these algorithms [11].

2.1. Deep Learning in Medical Imaging

The first notable study to apply deep neural networks to biomedical image processing was [4], which used a CNN architecture to perform pixel-wise classification of electron microscopy neuron images into membrane and non-membrane pixels. Due to the early successes of [4] and others, interest in applying CNN architectures to medical images has burgeoned in recent years [11].

Medical image analysis and segmentation problems present several unique challenges. First, patient data in medical imaging problems tends to be exceedingly heterogeneous [15], where the same pathology can present in very different ways across patients. Further complicating the challenge of medical image segmentation is the relatively small size of the data sets available, and the available data being incomplete or inconsistent. While most computer vision data sets such as [6, 5, 23] contain thousands or even millions of examples, in medical imaging problems there are rarely more than a few hundred examples in a data set; consequently, CNNs trained on these data sets are highly prone to overfitting [10]. Nevertheless, CNN-based methods have been shown to perform at least as well as other methods (e.g. support vector machine, generative models), and are very promising for applications in medical image segmentation [15].

2.2. Image Segmentation

There exist two main approaches to semantic segmentation: pixel-wise segmentation, where a small patch of an image is used to classify the center pixel, and fully-convolutional architectures as first proposed by [14], where the network input is the full image and output is a semantic segmentation volume. [17] and [1] have explored the latter using VGG-inspired [26] architectures and shown fully convolutional networks to have accuracy comparable to pixel-wise approaches with a significantly lower computational cost.

Several CNN-based methods have been proposed for brain tumor segmentation from multimodal MRI, including those based on segmenting individual MRI slices [8], volumetric segmentation [2], and CNNs combined with other statistical methods [10]. Nearly all current architectures for brain tumor segmentation use a pixel-wise U-net approach as in [3, 22], which have been promising but still show limited success. Furthermore, while [16] has applied fully convolutional networks to other biomedical problems, no study thus far has used a fully convolutional approach for the specific problem of brain tumor segmentation.

2.3. Transfer Learning

Transfer learning has been applied widely to CNNs in many visual recognition tasks to obtain high performance when minimal labeled data is available. [18] has shown that CNNs consistently learn similar image features even when trained on data sets that differ in statistics and tasks. Indeed, many state of the art results and networks use pre-trained CNN trunks such as [13, 26, 29, 7, 21] for efficient image feature extraction.

In medical image analysis, transfer learning is both especially useful and exceptionally challenging. Large-scale medical image datasets are very rare and often impractical to obtain, so transfer learning provides a viable means to train data-hungry neural networks. At the same time thought, medical image data sets tend to vary greatly depending on the patients and tasks, making transfer learning more difficult and nuanced than in other visual recognition tasks.

3. Model Architecture

Motivated by the tremendous success of deep neural networks in a variety of visual recognition tasks [13, 26, 29, 7, 21] such as image classification [23, 5] and semantic segmentation [14], in this section we propose three novel architectures for brain tumor segmentation from multimodal MRI: a baseline voxel-wise CNN, a fully-convolutional patch-wise CNN, and a full-image fully-convolutional CNN.

3.1. Preliminaries

We represent the input as a fourth order tensor $I \in n^3 \times C$, where the image is dimension n^3 with C channels per voxel. For this study, we will always take $C = 4$, since our

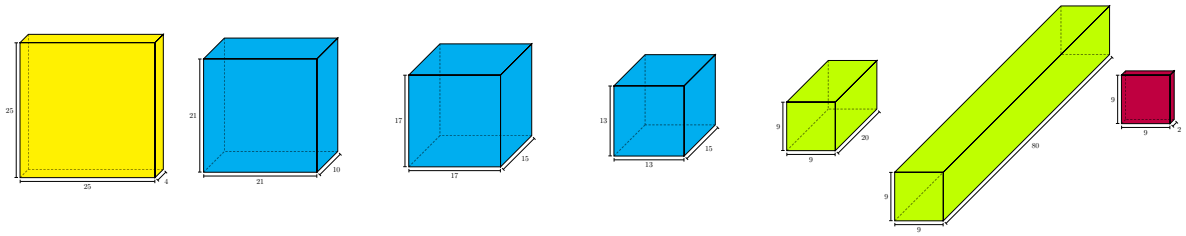


Figure 2: Baseline Architecture Diagram. Input is $25^3 \times 4$ volume (single slice shown, channels represented by slice depth). 3D convolutional layers (cyan) followed by ReLU activation. Fully connected layers (green) implemented as 1^3-kernel convolutions; first layer is followed by ReLU and dropout. Scores (red) for background and foreground.

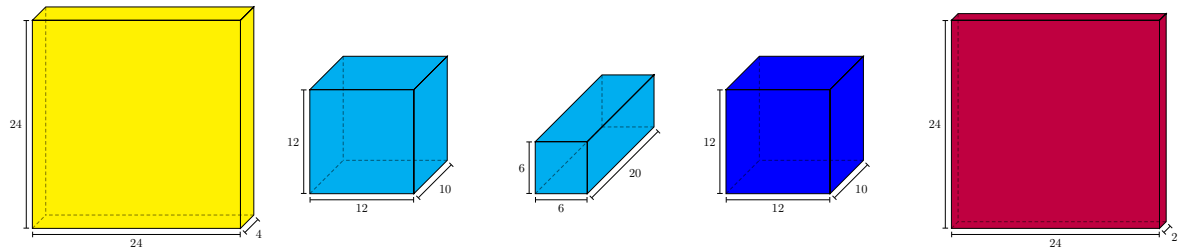


Figure 3: Patch-wise Fully-Convolutional Network (FCN) Architecture Diagram. Input same as baseline model. 3D convolutional layers (cyan) followed by ReLU activation, dropout, and 2×2 max-pooling. 3D deconvolutional layers (blue) followed by ReLU and dropout. Scores (red) for background and foreground for entire patch.

inputs contain 4 MR modalities. The output tensor O will have dimensions $O \in n^3 \times K$, where K is the number of semantic classes treated by the architecture (here $K = 2$).

3.2. Baseline Convolutional Network

The baseline convolutional network (BCN) model [28] functions by computing voxel-wise probabilities for each sub-region of the image. To do this, we use a CNN architecture with 3D convolutional kernels outlined in Fig. 2. Each convolutional layer is followed by a ReLU nonlinearity, which we omit in the diagram for brevity. The final two layers are fully-connected layers implemented as 1^3-kernel convolutions. The input to the baseline model is a $25^3 \times 4$ sub-cube randomly sampled from the input image tensor I, and the output is a 9^3 volume providing estimates for a 9×9 sub-patch of the input volume. We train the model using softmax cross-entropy loss with L_2 regularization and Adam optimization. The model architecture is shown in Fig. 2.

3.3. Fully-Convolutional Network

We also implement a patch-wise fully-convolutional network (FCN). The FCN takes as input a 24^3 patch of the input volume and outputs a 24^3 segmented volume. The network consists of two 3D convolution and two 3D deconvolution layers. The convolutions are followed by a ReLU activation function, dropout, batch normalization, and 2×2 max-pooling. The deconvolution layers are followed by ReLU activations and batch-normalization. Batch normalization has been shown in [14] to reduce internal covariance shift in convolution-deconvolution architectures, and dropout is well-known to help prevent overfitting during training and improve model generalization when evaluating on the test dataset. The model uses softmax cross-entropy loss with L_2 regularization and is trained using Adam optimization. The model architecture is shown in Fig. 3.

3.4. Full-image Fully-Convolutional Network

Finally, we propose a full-image fully-convolutional network (FIFCN) architecture inspired by that used in [17]. The deconvolutional network is a mirror of the forward network, with maxpooling layers replaced with unpool/deconvolution layers with learnable filters as originally proposed in [32].

The architecture in [17] was based on VGG16 [26], but we instead use a network loosely-based on the 11-layer CNN from [27]. To reduce the number of parameters, we first reduce the number of layers from 8 convolutional and 3 fully connected layers to only 5 convolutional layers. Second, we use $\sim 4\times$ less filters at each layer. We retain the ReLU activation function and 2×2 max-pooling after each convolutional layer.

To reduce the bias towards the background class and decrease computational cost, we first reduce the input volume

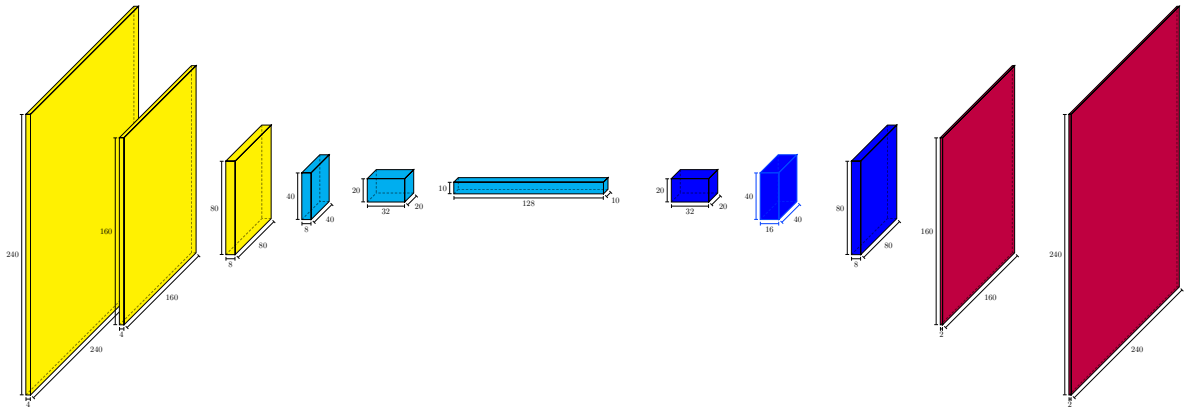


Figure 4: Full-Image FCN Architecture (FIFCN) Diagram. Input is full image (single slice shown, channels represented by slice depth). 3D convolutional layers (cyan) followed by ReLU activation and 2×2 max pooling. 3D deconvolutional layers followed by ReLU and batch normalization. Output (red) contains class scores for background and four tumor regions. Primary advantage of FIFCN is potential speedup in throughput.

size from $240 \times 240 \times 155$ to $160 \times 160 \times 144$, then further down sample via 2×2 max-pooling. The image then passes through the convolution and deconvolution layers. The output of the network is a $160 \times 160 \times 144$ volume with predictions for tumor and background classes. This volume is then padded with pixels labeled as background to create the final output volume. For the loss function, we examine the performance of both softmax cross-entropy loss (per [17]), and Dice score loss as in [16]. We include L_2 regularization and train using Adam optimization. FIFCN architecture is shown in Fig. 4.

4. Experiments

4.1. Dataset Overview

The BraTS 2017 dataset contains T1, T1 contrast enhanced, T2, and FLAIR images for a total of 243 patients (135 glioblastoma and 108 lower grade glioma) that have been manually segmented into tumor core, enhancing tumor, and background regions [15]. Example images are shown in Fig. 1. The BraTS dataset contains segmentations for necrotic core tumor, enhancing core tumor, non-enhancing core tumor, and edema regions. The full BraTS Challenge is to obtain the highest possible segmentation score for all four regions, but here we focus exclusively on segmenting tumor regions from the background.

We use 178 images for training the models and 44 for evaluation. To pre-process the input images, we normalize the images to be zero-mean and unit standard deviation. The images in the BraTS dataset have a consistent shape of $240 \times 240 \times 155$ voxels, so it is unnecessary to zero-pad or otherwise further process the images.

4.2. Performance Evaluation

For all tasks on both models, the primary performance metric is the mean dice score, which is a standard evaluation metric in the medical imaging and computer vision communities. The dice score for output predictions $P \in \{0, 1\}$ and the expert's consensus ground truth $T \in \{0, 1\}$ is defined as:

$$\text{Dice Score} = \frac{2|P_1 \cap T_1|}{|P_1| + |T_1|} \tag{1}$$

where P_1 and T_1 represent the set of voxels where $P = 1$ and $T = 1$ [15].

4.3. Results

The validation dice score results for all models trained on the BraTS dataset are summarized in Fig. 5, and a comparison with previous studies is shown in Table 1. The dice score data shows that the patch-wise approaches (BCN and FCN) perform dramatically better than the FIFCN models.

The poor performance of the FIFCN is likely due to several factors. First, because of the overwhelming number of "background" class pixels in comparison to the number of "tumor" class pixels in all training and test samples, the FIFCN model biases very heavily towards the background class. Secondly, the tumor is mainly distinguished from the background class by textural features as opposed to intensity values. Since the intensity is fairly homogeneous across the cortex (including tumor and edema regions), it is necessary for the model to learn textural features with which to represent and segment the image. This is challenging due to the relatively small size of the training set and comparatively large number of parameters in the model, so the

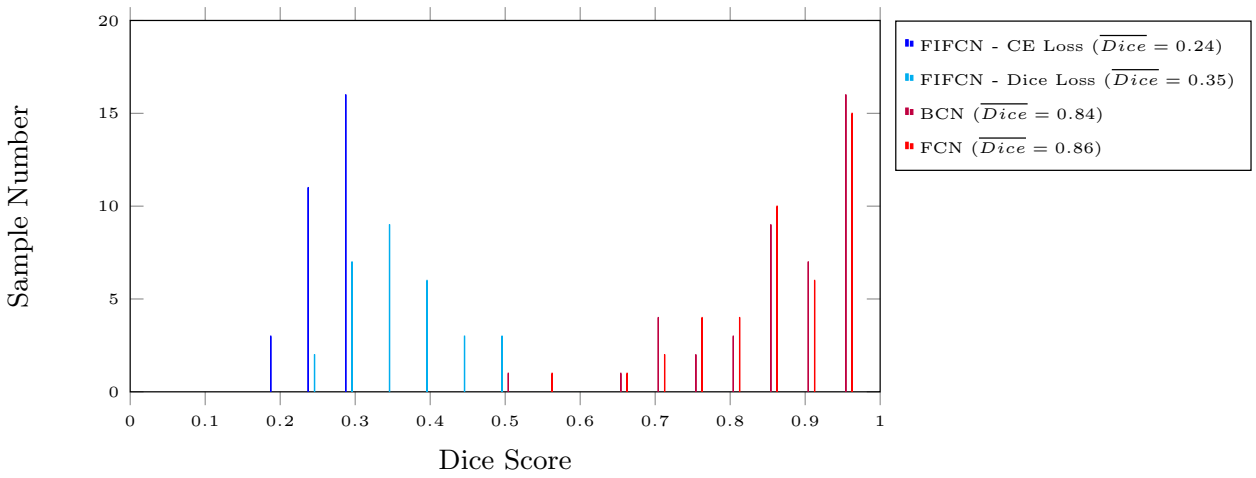
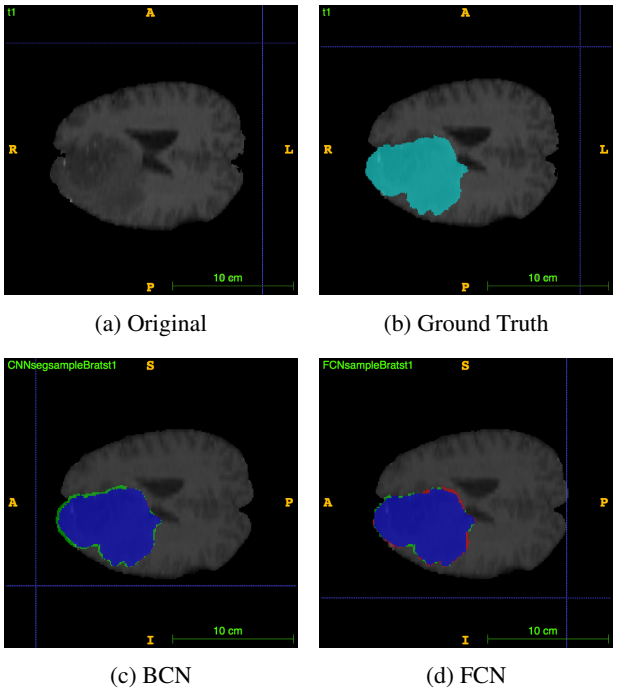


Figure 5: Histogram of dice scores across validation set for models trained and tested on BraTS dataset. The BCN and FCN models (mean dice 0.84, 0.86) performs strongly compared to the top benchmark accuracy (mean dice 0.90 [15]). The FIFCN models do not perform very well. The dice score loss model performs better, but the scores are poor compared to the patch-wise architectures. This is most likely due to the FIFCN model biasing heavily towards the background class.



(a) Original (b) Ground Truth

(c) BCN (d) FCN

Figure 6: (a) Original T1 image, (b) Ground truth segmentation (cyan), (c) Segmentation via BCN model, (d) Segmentation via FCN model. Correct voxels (blue), unidentified voxels (red), mis-identified voxels (green). The segmentation quality from both models is comparable. Images from ITK-Snap.

model seems to be overcome with bias towards background pixels before it is able to learn useful textural features.

The patch-wise models perform much more strongly since a patch will typically contain a majority of tumor or non-tumor pixels. During training, the BCN and FCN are trained on majority-tumor patches and majority-background patches which randomly selected with equal probability. Consequently, an implicit classification between tumor and non-tumor patches becomes built into the model, thereby increasing the segmentation accuracy.

An example of brain MRI slices from validation set images segmented via the BCN and FCN models are shown in Fig. 6. We observe that the patch-wise segmentation does not disproportionately over- or underestimate the number of pixels in the tumor region, and qualitatively provides very strong segmentation results.

While overall the FIFCN model does not perform well, the dice score loss performances significantly better than the cross entropy loss when trained on the same model and dataset. This is in line with the results in [16], which showed a significant improvement in accuracy for brain tumor segmentation when the network was trained to maximize the dice score instead of other more conventional loss functions (e.g. cross-entropy loss). Therefore, these results validate the findings of [16] and support the use of a dice score loss for medical image segmentation.

The patch-wise approaches could be improved most easily by including pre-computed textural features (e.g. Gabor wavelets) in the model, or by employing a U-Net approach as in [16]. Similarly, the FIFCN could be improved by using a U-Net approach, since this would enable more high-level image features to be preserved. In general, the performance of the FIFCN seems to suffer primarily due to the strong bias towards background pixels, so future iterations should

include a structural change to address this bias.

4.4. Ablation Study

We also perform an ablation study for the BCN and FCN models. The results from the ablation study is summarized in Table 1. In general, adding dropout and batch normalization layers noticeably increases the dice score for the models. Example loss curves and dice scores are given in 7. We observe that the batch normalization and dropout model has a significantly lower final loss score, but the mean dice score is not greatly improved over the other models with dropout and batch normalization. Overall, the most accurate versions of the two models perform comparably to different architectures presented at the 2014 MICCAI BraTS Challenge workshop. It should be noted that these studies were evaluated on the 2013 and 2014 iterations of the BraTS dataset and were for the full dataset i.e. both HGG and LGG samples, whereas our model is evaluated on only HGG patients with the 2016 and 2017 datasets.

5. Transfer Learning

5.1. Dataset

The Rembrandt dataset from [25] contains 57 multimodal images of glioma patients. The Rembrandt dataset is significantly smaller than the BraTS dataset, but has the same task and therefore naturally lends itself to transfer learning via the BraTS dataset.

The primary differences between the BraTS and Rembrandt datasets are the image resolution and the acquisition sequences. The Rembrandt images are $160 \times 160 \times 53$ voxels, which is smaller than the BraTS images, and were also acquired with different MR pulse sequences than the BraTS dataset. While a given MR modality will have the same general intensity and textural features regardless of the pulse sequence, different sequences can cause differences in image quality.

The approach we take for transfer learning is to pre-train a model on the larger BraTS dataset, then use this as an initialization for the Rembrandt segmentation network and train on the Rembrandt dataset. We focus on applying transfer learning with the BCN and FCN models since these were the strongest architectures for the BraTS dataset. We present results for the "original" architecture, which has not been modified from that used on the BraTS data, and "fine-tuned" architectures, which were pretrained on the BraTS data but modified to deliver stronger results for the Rembrandt images.

5.2. Model Structure

For transfer learning, we employ the same general architectures as for the BraTS dataset. In the "original" architectures, we do not modify the model beyond accommodating for the different input and output size required by the Rembrandt dataset. In the "fine-tuned" architectures, we add additional dropout and batch normalization layers to improve performance. Results for both architectures are presented in the following section.

5.3. Experiments

We initially train on the BraTS dataset, then further train the model using 46 images from the Rembrandt dataset, saving 10 images as the validation set. Fig. 8 summarizes the segmentation results for the transfer learning. There are three main takeaways from the results. First, we observe that fine-tuning the model significantly improves the segmentation quality, both in terms of the histogram (the distribution is much more skewed left for the fine-tuned models) and the mean dice score. This change is expected since the original model was designed to work well with the BraTS dataset, whereas the design choices for the fine-tuned model is made based on Rembrandt images.

Secondly, we observe that the BCN slightly outperforms the FCN. This is most likely due to the BCN relying less on the specific image features than does the FCN. In the FCN, the convolutional layers compress the image as high-level features, then the deconvolutional layers rebuild the segmented image from these features. While this is a potentially very powerful architecture, the implicit assumption is that high level features for similar images will also be similar. However, the difference in quality and resolution of the Rembrandt images compared to the Brats images means this assumption may not hold very strongly, and therefore we see a drop-off in segmentation quality between the BraTS and Rembrandt datasets. A similar effect of the high-level features being corrupted by the differences between the images can also account for the sharp decrease in performance of the BCN model when moving from the BraTS to the Rembrandt dataset, but some of this drop-off is presumably tempered by the use of fully-connected layers instead of deconvolutional layers to perform the prediction.

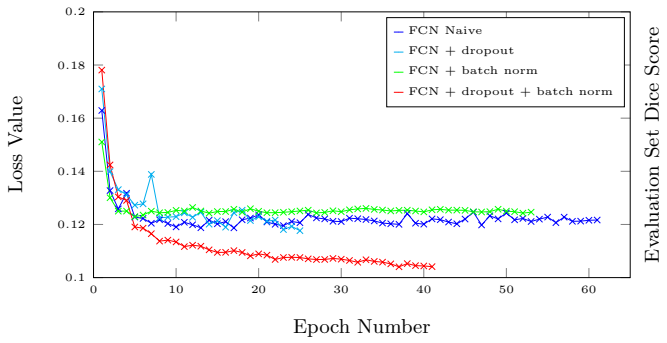
Finally, the results show that the segmentation quality is very inconsistent across the validation set. For all models, we observe both very high and very low segmentation quality. The fine-tuned FCN model in particular has samples in every bin across the histogram. This shows that while transfer learning may be promising for applications to glioma segmentation, it is not necessarily reliable and is still highly dependent on image quality and resolution.

Fig. 9 shows an example Rembrandt image segmented via the FCN model. Qualitatively, fine-tuning the model noticeably improves the segmentation quality, as shown by the results in Table 2. In particular, the number of unidentified voxels (red) is significantly reduced using the fine-tuned model. However, the segmentation quality is still poorer than for the BraTS dataset.

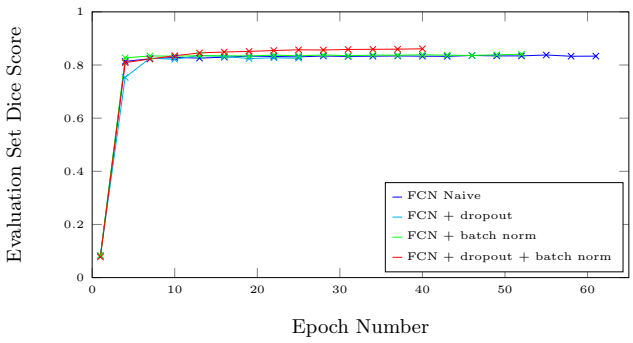
Study	Method	Mean Dice Score (%)
Urban [19]	Deep CNNs	87
Reza [19]	Random forest	92
Goetz [19]	Randomized trees	83

Model	Version	Average Dice Score (%)
BCN	—	82.7
BCN	+ Dropout	84.2
BCN	+ Batch Normalization	83.4
BCN	+ Dropout + Batch Normalization	**84.5**
FCN	—	83.8
FCN	+ Dropout	83.6
FCN	+ Batch Normalization	84.1
FCN	+ Dropout + Batch Normalization	**86.1**

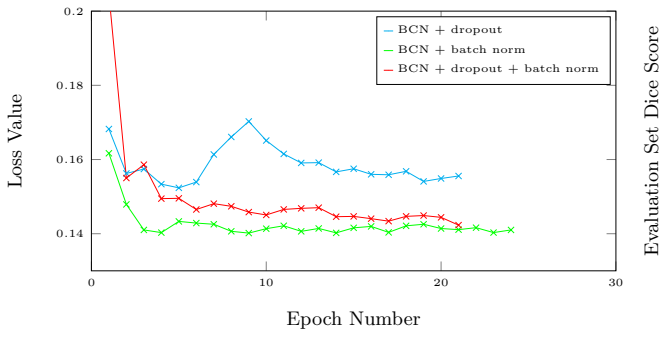
Table 1: Ablation study and comparison with previous BraTS Challenge studies. Note that the previous studies used both HGG and LGG patients from an earlier iteration of the BraTS dataset for training and evaluation. Here, we focus on HGG patients with the most current BraTS dataset.



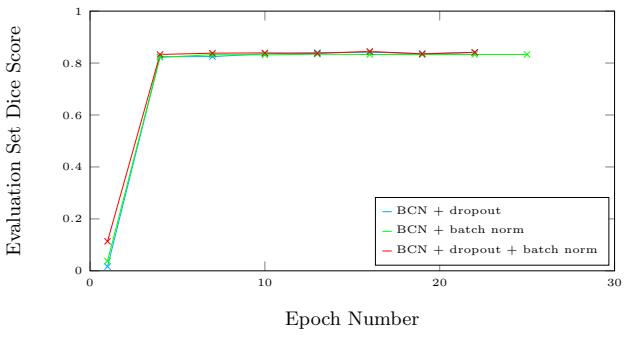
(a) FCN Loss curves



(b) FCN Dice score curves



(c) BCN Loss curves



(d) BCN Dice score curves

Figure 7: Loss and evaluation set dice score curves for the ablation study. Here, the dice score was evaluated on the evaluation set every 3 iterations during training.

Overall, transfer learning is a promising approach for augmenting and initializing CNNs for brain MRI segmentation when using small datasets, but there remains much work to be done. Specifically, there are many challenges to overcome in pre-training and training for images of differing resolutions. However, as the fine-tuning results demonstrates, it is indeed possible to significantly increase the accuracy of a model by fine-tuning, and this in turn suggests that with further work it will be possible to effectively apply transfer learning to brain MRI segmentation.

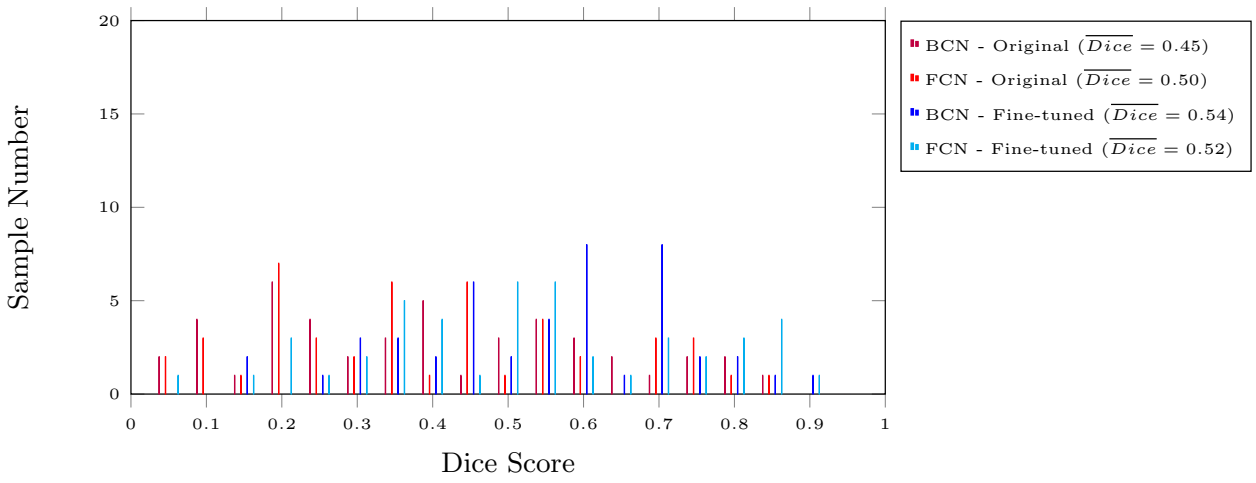
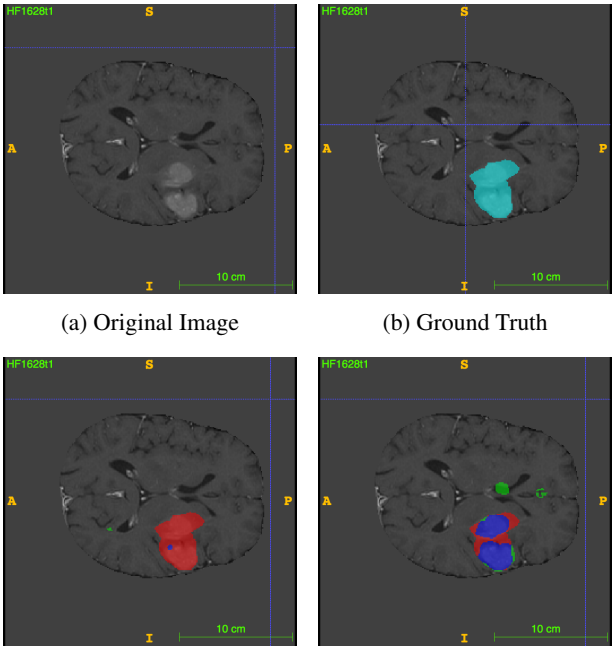


Figure 8: Histogram of dice scores across validation set. Fine-tuning the model by adding batch-normalization layers increases the mean accuracy, and shifts the score distribution to being more skewed left. There are also more samples with high dice scores (i.e. > 0.85) with the fine-tuned models. Overall, the dice scores are inconsistent across the validation set.



(a) Original Image (b) Ground Truth

(c) FCN Before Fine-tune (d) FCN Fine-tuned

Figure 9: Examples of segmentation via transfer learning. (a) Original image, (b) Ground truth segmentation (cyan), (c) FCN segmentation before fine tuning, (d) FCN model segmentation after fine-tuning. Correct pixels (blue), unidentified voxels (red), mis-identified voxels (green). We see that fine-tuning the model noticeably improves the segmentation quality. Images from ITK-Snap.

Model	Method	Average Dice Score (%)
BCN	Pre-trained	40.8
BCN	Re-trained	45.2
BCN	Fine-tuned	**53.5**
FCN	Pre-trained	38.2
FCN	Re-trained	50.3
FCN	Fine-tuned	**52.2**

Table 2: Ablation study and comparison for transfer learning experiments of both BCN and FCN. We evaluate on the same Rembrandt dataset with the model pre-trained on Brats, the model retrained on a small Rembrandt training subset (11 samples), and the model fine-tuned and retrained on the training subset.

6. Conclusion

Here we have developed three novel architectures for brain tumor segmentation and evaluated their accuracy on the BraTS Challenge 2017 dataset, and also explored the application of transfer learning from the BraTS architecture to the Rembrandt dataset. Many of the results presented here—specifically, the use of patch-wise approaches to glioma segmentation—are very promising. Future work should focus on developing a more complex FCN architecture, and applying dice loss to both the BraTS dataset and transfer learning. Overall, we are optimistic that with further work, it will be possible to use CNN-based architectures to efficiently and effectively segment brain tumors, and thereby bring many of the applications such as surgical planning and visualization closer into reach.

7. Acknowledgements

1) Prof. Olivier Gevaert introduced and instructed this brain tumor segmentation problem. Raghav Subramaniam in Prof. Gevaert's lab provided the base framework for the naive BCN model presented here.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. pages 1–14, 2015. 2

[2] H. Chen, Q. Dou, L. Yu, and P.-A. Heng. VoxResNet: Deep Voxelwise Residual Networks for Volumetric Brain Segmentation. pages 1–9, 2016. 2

[3] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-net: Learning dense volumetric segmentation from sparse annotation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9901 LNCS:424–432, 2016. 2

[4] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. *Nips*, pages 1–9, 2012. 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: a large-scale hierarchical image database. In *CVPR*, 2009. 2

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2

[7] R. Girshick. Fast r-cnn. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[8] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain Tumor Segmentation with Deep Neural Networks. 2015. 2

[9] E. C. Holland. Progenitor cells and glioma formation. *Current Opinion in Neurology*, 14(6):683–688, 2001. 1

[10] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61–78, 2017. 1, 2

[11] B. Kayalibay, G. Jensen, and P. van der Smagt. CNN-based Segmentation of Medical Imaging Data. 2017. 2

[12] B. A. Kohler, E. Ward, B. J. McCarthy, M. J. Schymura, L. A. G. Ries, C. Eheman, A. Jemal, R. N. Anderson, U. A. Ajani, and B. K. Edwards. Annual report to the nation on the status of cancer, 1975-2007, featuring tumors of the brain and other nervous system. *Journal of the National Cancer Institute*, 103(9):714–736, 2011. 1

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 2

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR (to appear)*, Nov. 2015. 2, 3

[15] B. H. Menze et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015. 1, 2, 4, 5

[16] F. Milletari, N. Navab, and S. A. Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 565–571, 2016. 2, 4, 5

[17] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, 11-18-December-2015:1520–1528, 2016. 2, 3, 4

[18] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[19] M. . Proceedings. Miccai 2014. pages 1–39, 2014. 7

[20] S. Ray, M. M. Bonafede, and N. A. Mohile. Treatment patterns, survival, and healthcare costs of patients with malignant gliomas in a large US commercially insured population. *American Health and Drug Benefits*, 7(3):140–149, 2014. 1

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[22] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. pages 1–8, 2015. 2

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 2

[24] L. Scarpace, A. E. Flanders, R. Jain, T. Mikkelsen, and D. W. Andrews. Data from rembrandt, 2015. 2

[25] L. Scarpace, A. E. Flanders, R. Jain, T. Mikkelsen, and D. W. Andrews. Data from rembrandt, 2015. 6

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 3

[27] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICRL)*, pages 1–14, 2015. 3

[28] R. Subramaniam. tumor-seg. https://github.com/raghavsub/tumor-seg, 2016. 3

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015. 2

[30] K. Urbanska, J. Sokolowska, M. Szmidt, and P. Sysa. Glioblastoma multiforme - An overview. *Wspolczesna Onkologia*, 18(5):307–312, 2014. 1

[31] P. Y. Wen, D. R. Macdonald, D. A. Reardon, T. F. Cloughesy, A. G. Sorensen, E. Galanis, J. DeGroot, W. Wick, M. R. Gilbert, A. B. Lassman, C. Tsien, T. Mikkelsen, E. T. Wong, M. C. Chamberlain, R. Stupp, K. R. Lamborn, M. A. Vogelbaum, M. J. Van Den Bent, and S. M. Chang. Updated response assessment criteria for high-grade gliomas: Response assessment in neuro-oncology working group. *Journal of Clinical Oncology*, 28(11):1963–1972, 2010. 1

[32] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013. *Computer VisionECCV 2014*, 8689:818–833, 2014. 3