CIS 5528 BraTS Brain Tumor Detection

Andrew Gnias, Tarius Hill
Temple University
Philadelphia, PA
{andy.gnias,tarius.hill}@temple.edu

Abstract

Image segmentation techniques have been used extensively in the field of biomedicine, including in the detection of brain tumors. While many brain tumors are benign, those that are malignant pose a 5-year survival rate of only 35.6%, making accurate brain tumor detection particularly important. This paper investigated the use of autoencoder neural networks and clustering methods to detect and segment brain tumors from MRI scans as part of the Brain Tumor Segmentation (BraTS) Continuous Challenge. At this point in our research, we have been able to detect brain tumors in MRI scans with an F-score of 0.70 using a Residual U-Net neural network. The paper also offers suggestions for improvement, and future work that may occur outside the scope of the class.

***K*eywords** cancer · brain tumor · computer vision · deep learning · image segmentation · brats

1 Introduction

A tumor is defined as any abnormal cell mass that occurs due to uncontrolled cell division[1]. Tumors can either be benign or malignant. Benign tumors do not invade neighboring tissues or organs, whereas malignant tumors can invade surrounding tissues. Malignant tumors grow more quickly than benign ones and have no specific shape, making classification and accurate diagnosis difficult[2].

Brain tumors are among the most common tumors within the central nervous system[2]. While they require less invasive methods to diagnose and monitor the response to treatment, they are often recognized too late to conduct a successful course of treatment plan[3]. Therefore, examining different detection methods and implementing machine learning within the field of oncology can serve significant benefits.

Modern techniques and procedures for diagnostic imaging include CT and PET scans, regular and advanced MRIs, Diagnostic Angiograms, and Myelograms[4]. Magnetic Resonance Imaging (MRI)

is the main method through which brain scans are captured. Traditionally, these scans would be manually observed by a physician to determine if a brain tumor is present. However, the increase in computing power and machine learning methods over the past 10 years has allowed for more effective deep learning methods to be used in the field of biomedicine. Most notably, deep neural networks have significantly contributed to the process of medical image segmentation.

Image segmentation is the process of dividing images into subgroups to reduce its complexity and make meaningful inferences from these subgroups[5]. In the field of biomedicine, image segmentation is often used to detect abnormalities in medical images, such as detecting a tumor in an MRI. To do this, the MRI first needs to be converted into a machine readable format, such as the NIfTI (Neuroimaging Informatics Technology Initiative) format. This allows an MRI scan to be converted into a 3-dimensional matrix where each index represents the pixel intensity of the image. This 3-dimensional matrix can then be used as an input to an image segmentation model which will then analyze the image.

The current gold-standards for medical image segmentation are clustering methods and deep neural networks. Clustering is an unsupervised machine learning method that groups sample data into similar groups[6]. In the scope of tumor identification, clustering attempts to identify a cluster that segments the tumor from the rest of the brain. Neural networks are algorithms that mimic the function of the human brain, using several input and output layers to fine tune an output that provides meaningful inferences about the input data[7]. In this context, the neural network acts as a supervised model where it learns features from training data, and applies a segmentation to validation images which attempts to encompass the entirety of the tumor.

In this project, our group will use clustering and deep neural networks to identify tumors from 3-dimensional MRI data. This data is being provided to our group as part of the Synapse 2021 Brain Tumor Segmentation (BraTS) Challenge. For clustering, we will apply k-means clustering to the dataset using pre-existing libraries in R. For deep neural networks, we will focus on using pre-built versions of Residual U-Net, an auto-encoder network, and SegResNet, a segmentation network with residual blocks, from the Python Monai library. These will be used to generate segmentation layers that encompass the area of the MRI containing a tumor.

2 Related Work

The concept of using neural networks for medical image segmentation is not new. U-Net, an auto-encoder network proposed in the 2015 paper, "U-Net: Convolutional Networks for Biomedical Image Segmentation", was designed specifically for medical image segmentation. The model was successfully used in segmentation tasks for neuronal structure recordings taken from an electron microscope and cell images taken from a light microscope[8]. This work has also inspired improved derivations of the U-Net architecture, such as Residual U-Net, which improved upon U-Net with residual blocks and was used to measure the diameter of the heart's left ventricle[9]. In 2018, developers at Nvida created a similar algorithm to U-Net, now commonly referred to as "SegResNet"[10]. The algorithm was used to win first place in the 2018 BraTS competition, the

same competition that this project is based on[11]. ResGANet, developed in 2021 with the hopes of improving the accuracy of medical segmentation tasks, uses a group attention network to build off the success of ResNet while also reducing the number of the model's overall parameters compared to ResNet[12].

Clustering methods have also been used for medical image segmentation tasks. While considered less effective than a deep neural network, clustering methods are appealing because they tend to be easy to use and require less computing power[13]. In "Brain Tumor Detection Using Color-Based K-Means Clustering Segmentation", researchers used k-means clustering to segment a tumor from an MRI brain scan.[13]. While this result is promising, it was only performed on 1 scan, so it does not clearly demonstrate that k-means is applicable to a larger dataset. "Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm" demonstrated more promising results, segmenting 50 2-D MRI images with accuracy ranging from 85-95%[14]. While this research utilized k-means clustering, it also combined the method with a modified watershed algorithm, demonstrating the potential need to augment clustering to successfully segment medical image data.

3 Methodology

3.1 Neural Network

3.1.1 Data Preparation

The data provided by the BraTS challenge is in the NIfTI format. The dataset includes 1251 MRI scans with tumor segmentation labels. Of these scans, 70% were used for training the models, 15% were used for validation within each training epoch, and 15% were used for testing once training was complete. Each MRI scan contained 4 MRI sequence types: T1, T1-weighted, T2-weighted, and T2-FLAIR, each representing different MRI imaging methods[15]. Each MRI scan also contained a segmentation scan that represented the tumor with 3 different tumor subclass labels: GD-enhancing tumor, invaded tissue, and tumor core[16]. Each image and segmentation is displayed on the transverse plane of the body with 150 frames. Two different methods were used for generating a tumor segmentation model with this data, and those methods are described below.

Single-Channel Image, Binary Classification Label (SCBC) Initial trials utilized a single-channel MRI scan where the T2-FLAIR MRI scan was used as the input image. For the segmentation label, all tumor subclasses were combined into 1 label, where 0 meant that a pixel in the image did not contain a tumor and 1 meant that a tumor was present. This method will be referred to as the SCBC (Single-Channel, Binary Classification) method.

Multi-Channel Image, Multi-class Label (MCMC) Subsequent trials used a 4-channel MRI scan where each available MRI scan type was used as a channel in the input image. Additionally, all available scan types were used and the tumor sublcasses in the label were retained. This transformed the problem from a binary classification problem that determined if a tumor was present at each

pixel in the MRI scan to a multi-class classification problem that determined not only if a tumor was present, but which portion of the tumor was present. This method will be referred to as the MCMC (Multi-channel, Multi-class) method.

3.1.2 Neural Network: Design

Two neural networks will be used on the dataset: Residual U-Net and SegResNet.

Residual U-Net Residual U-Net is an auto-encoder neural network that improves upon U-Net. In the encoding path, the network repeatedly applies 2 3x3 convolutions and the Parametric Rectified Linear Unit (ReLU) activation function to the data. It then applies a max pooling operation, which assists in removing unhelpful features from the data[17]. The decoding portion of the network also uses 3x3 convolutions with Parametric ReLU, but instead of max pooling it performs a 2x2 up-sampling convolution to restore the image to its original size. The network then applies a 1x1 convolution to map pixels to output classes[8]. Our project will use the Monai implementation of Residual U-Net with 5 encoding and decoding layers. The Adam optimizer will be used with a learning rate of 0.015.

SegResNet SegResNet is also an auto-encoder neural network, but uses a larger encoder network than its decoder. The encoder sequence uses ResNet blocks with 2 normalized convolutions and ReLU as the activation function. In each layer, the data is downsampled by a factor of 2 while the feature size increases by a factor of 2. In the decoder, the reverse occurs where features are reduced by a factor of 2 while the data is upsampled to its original size[11]. The Monai implementation of SegResNet will be used with 4 downsampling layers and 3 upsampling layers[18]. The Adam optimizer will be used with a learning rate of 0.0065.

3.1.3 Training and Validation

Because the neural network architectures from the Monai library are implemented in PyTorch, other commonly used machine learning functions and processes from PyTorch can be used to train and validate our models.

Training Loop The overall process for training the model for each batch is as follows:

- Apply Transform functions to the batch
  - Convert NIfTI image into 3D tensor matrix
  - Swap matrix indices so channel length is first
  - Prepare Segmentation Label
    - For the SCBC method, combine segmentation labels into 1 tumor label, where 0 indicates that the tumor is not present at 1 indicates that the tumor is present
    - For the MCMC method, Convert label to a One-Hot label format, where each index is a length 3 array indicating whether each tumor subclass is present or not

  - Update the image orientation for a 3D image
  - Scale the pixel intensity from 0-255 to 0-1
  - For the MCMC method, create a multi-channel image with all available modes of the MRI scan
  - Resize images from 240x240x155 to 128x128x64
- Zero out the gradient
- Pass the image batch into the model
- Calculate the loss between the model output and the segmentation label using Dice Cross-entropy loss
- Recalculate the model weights via propagation
- Update model parameters via the optimizer

Validation Loop The process for epoch validation and testing a completed model are nearly identical to each other and follow the following steps for each batch:

- For testing, load the best model weights from training. For training, use the current state of the model as it is being trained
- Apply Transform functions to the batch (identical to transforms used in the training loop)
- Call model on image batch using a sliding window inference
- Post-process results using Sigmoid Activation and Discretization functions
- Calculate efficacy using Mean Dice metric (F-score)
- For validation within an epoch, if the F-Score is better than all previous epochs, save the model weights

3.1.4 Hyperparameter Optimization

A Hyperparameter Optimization script was used to determine the ideal Optimizer and Learning Rate for the neural network. This was done utilizing Optuna, a Python hyperparameter optimization library which iterates through different training runs of the model with variable optimizer types and learning rate values. The script records the parameters used as well as the F-score of the model, allowing the parameters of the best performing model to be used in a training run. This was done for selecting the hyperparameters of the SCBC model. However, because of the long run time of running the script for the MCMC model, hyperparameter optimization was not performed, and the same hyperparameters as the SCBC model were used.

3.1.5 Hardware

The complexity of the networks being used and the density of the MRI data requires GPU processing for model training and validation to complete within a reasonable amount of time. The models will be trained on a privately owned Linux PC with a GeForce RTX 3060 Ti 8GB GPU. Training epochs

took approximately 5 minutes for the SCBC method and 20 minutes for the MCMC method. Using 150 epochs, training a model took around 12 hours for the SCBC method and over 2 days for the MCMC method.

3.2 Clustering

Clustering methods were also performed on the BraTS dataset. Because clustering is an unsupervised machine learning algorithm, less preparation was needed to preprocess the images for clustering. The T1, T2-FLAIR, and Segmentation channels from a single MRI image were used in 3 separate clustering runs. The scans were converted into a 3-dimensional matrix where each index represented the pixel intensity of the image. k-means clustering was applied to the preprocessed data using the "kmeans()" function in the "stats" package in R. To determine the optimal number of clusters, the "elbow method" was used, where the within-cluster sum of squares (WSS) was calculated for a range of cluster numbers (k=2 to k=10). The value of k that resulted in the largest decrease in WSS was selected as the optimal number of clusters, which in this case was $k = 3$.

4 Results

4.1 Neural Network: Single Channel Image with Binary Tumor Label (SCBC)

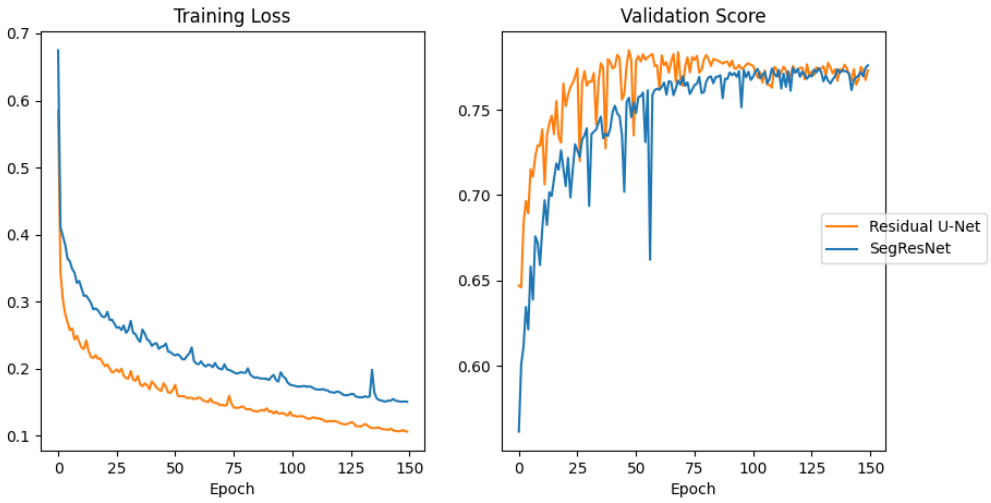

Figure 1: Graphs of Loss and Validation Metrics of each model over 150 epochs

Model	Lowest Loss	F-Score
Residual U-Net	0.1065	0.79
SegResNet	0.1508	0.78

Table 1: Loss and Validation Metrics for each model

4.2 Neural Network: Multi-Channel Image with a Multi-Class Tumor Label (MCMC)

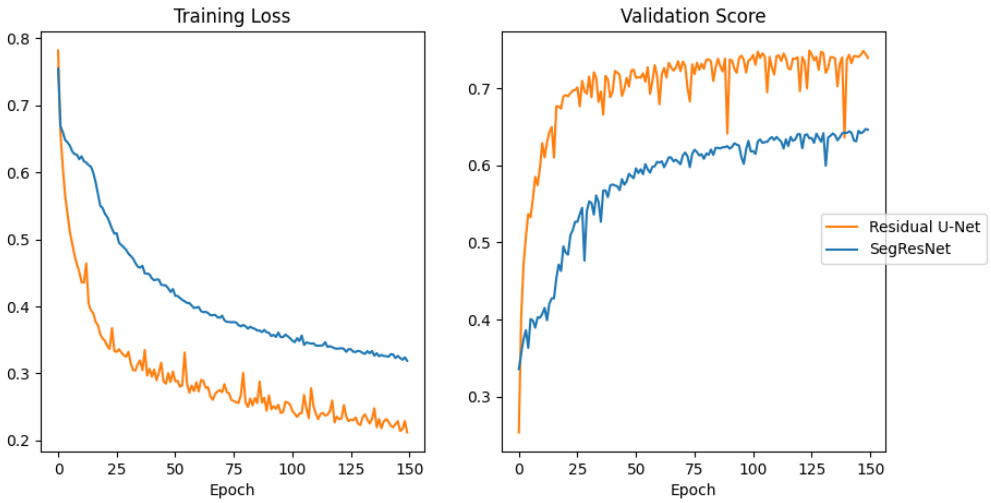

Figure 2: Graphs of Loss and Validation Metrics of each model over 150 epochs

Model	Lowest Loss	F-Score
Residual U-Net	0.1702	0.704
SegResNet	0.3190	0.644

Table 2: Loss and Validation Metrics for each model

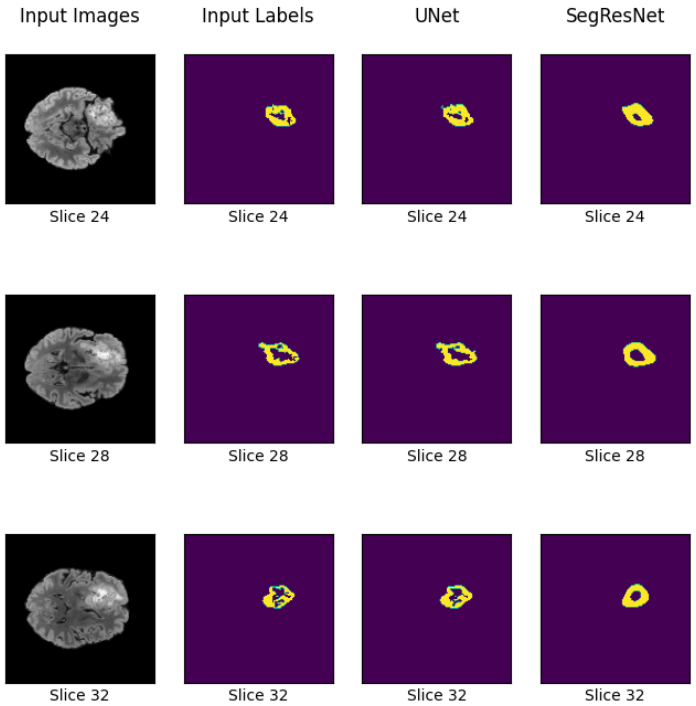

Figure 3: Visual Representation of Input MRI Scan, Expected Segmentation, and Model Segmentation over 3 selected frames of the MRI

4.3 Clustering

	T1	T2-FLAIR	Segmentation
centers	640.551, 0.565, 906.978	849.67, 3.95, 1318.73	0.00132, 4.00, 2.00
totss	8.38E+11	1.39E+12	583604.1
withinss	6.50e+09, 1.07e+09, 7.20e+09	1.82e+10, 8.56e+09, 2.19e+10	11722, 0, 0
tot.withinss	14774869944	48739823123	11722.49
betweenss	823490086705	1.34E+12	571881.6
size	537400, 7465601, 924999	763932, 7570995, 593073	8882433, 32731, 12836
iter	2	3	1
ifault	0	0	0

Table 3: Output from k-means for the T1, FLAIR, and Segmentation Channels of the MRI Scan

5 Discussion

5.1 Neural Network

Our results indicate modest success in segmenting brain tumors from the MRI images in the dataset. In the initial runs where a single-channel image was used with a binary classification label, we were able to achieve validation scores of 0.79 for a Residual U-Net network and 0.78 for a SegResNet network. After this initial success, we made the decision to add features to the input image by using a 4-channel image, and increase the complexity of the problem by attempting to not only detect if a tumor exists at a pixel, but which part of the tumor the pixel contained. Using this multi-channel image type with a multi-class label, we achieved validation scores of 0.70 for Residual U-Net and 0.64 for SegResNet. While it was clear the model was still learning and able to segment tumors to a certain degree, our performance was now worse. This was somewhat expected. Using a multi-class label and attempting to detect not only the presence of a tumor, but what part of the tumor was in a pixel made the problem more complex and thus the MRI scan harder to accurately segment. However, we had hoped that going from a 1 to 4-channel image would give the model more features to learn from, and this more complex input image would make up for any increased difficulty in solving the problem.

Compared to available literature, our model performed much worse than expected. "3D MRI brain tumor segmentation using autoencoder regularization", the academic paper presenting SegResNet, achieved between 0.76 and 0.88 Validation Mean Dice scores on different portions of a brain tumor from the BraTS dataset, whereas our version of SegResNet had a score of 0.64.

It is possible that practical optimizations to the existing model could improve its performance. The learning rates used for our Adam optimizer for both neural networks were determined from a hyperparameter optimization script. This script utilized a single channel image and a binary label. Rerunning this script with the multi-channel image, multi-class label could indicate that a better

learning rate and / or optimizer could be used for each model. However, because of the long run time of training, true hyperparameter optimization cannot be achieved. Assuming we want to iterate over different optimizers and learning rates, we would ideally like to have at least 10 runs of the hyperparameter optimization script to complete. If we limit the optimization to 50 epochs and assume each epoch would take 20 minutes to complete, it would take around 7 days for an adequate hyperparameter tuning session to complete.

5.2 Clustering

While our initial results showed some success in applying k-means clustering to MRI data, it was not sufficient or promising enough for further work. Raw cluster data was able to be generated, but we have been unable to apply these results to actual cluster groups on the 3D MRI scan. Without the ability to validate these results visually or quantitatively, clustering does not give us a method for classifying tumors in an MRI scan. As discussed in 2. Related Work, clustering has been used to segment brain tumors. However, due to the time constraints of the project, the inexperience of the group in clustering methods, and the proven success of the neural network, it was determined not to pursue clustering further.

5.3 Future Work

In improving these results, we can look at modifications that can be made to both the network itself and the data. For the network itself, the use of transfer learning models have the potential to show improvement over our current neural network implementation. Using a pre-trained model may allow us to benefit from a neural network trained on a diverse set of data compared to one trained on BraTS data alone.

For the data, it would be beneficial to test the model on an external data source to determine its applicability to all MRI scans, and not just those provided by the BraTS Challenge. Additionally, automated methods for tumor detection should not just detect where a tumor exists, but if a tumor exists at all. Each model could be improved by incorporating MRI scans without tumors, and verifying if the model is successful in not making a segmentation where no tumor exists.

The use of lesser known models such as ResGANet, briefly described in Section 2, also show promise in medical segmentation tasks. As of April 2023, no pre-built implementation of ResGANet has been implemented and open-sourced in Python, so this would require manually creating the neural network using PyTorch. However, at the very least, it offers an exciting opportunity for work on the BraTS challenge outside the scope of the class project.

6 Code

All code used for the development of the neural networks can be accessed at https://github.com/AGnias47/brats-challenge-cis-5528.

References

[1] "tumor." https://www.cancer.gov/publications/dictionaries/cancer-terms/def/tumor.

[2] A. Patel, "Benign vs malignant tumors." https://jamanetwork.com/journals/jamaoncology/fullarticle/2768634.

[3] P. A. McKinney, "Brain tumours: Incidence, survival, and aetiology," *Journal of Neurology, Neurosurgery &; Psychiatry*, vol. 75, no. suppl_2, p. ii12–ii17, 2004.

[4] "How we diagnose brain tumors." https://www.dana-farber.org/brain-tumors/diagnosis/.

[5] "Image segmentation: The basics and 5 key techniques." https://datagen.tech/guides/image-annotation/image-segmentation/.

[6] "2.3. clustering." https://scikit-learn.org/stable/modules/clustering.html.

[7] "What are neural networks?." https://www.ibm.com/topics/neural-networks.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv:1505.04597v1*, Mar 2015.

[9] E. Kerfoot, J. Clough, I. Oksuz, J. Lee, A. P. King, and J. A. Schnabel, "Left-ventricle quantification using residual u-net," *Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges*, p. 371–380, 2019.

[10] "Nvidia data scientists take top spots in miccai 2021 brain tumor segmentation challenge." https://developer.nvidia.com/blog/nvidia-data-scientists-take-top-spots-in-miccai-2021-brain-tumor-segmentation-challenge/, Aug 2022.

[11] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization." https://arxiv.org/abs/1810.11654, 2018.

[12] J. Cheng, S. Tian, L. Yu, C. Gao, X. Kang, X. Ma, W. Wu, S. Liu, and H. Lu, "Resganet: Residual group attention network for medical image classification and segmentation," *Medical Image Analysis*, vol. 76, p. 102313, 2022.

[13] M.-N. Wu, C.-C. Lin, and C.-C. Chang, "Brain tumor detection using color-based k-means clustering segmentation," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, vol. 2, pp. 245–250, 2007.

[14] H. Ng, S. Ong, K. Foong, P. Goh, and W. Nowinski, "Medical image segmentation using k-means clustering and improved watershed algorithm," in *2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 61–65, 2006.

[15] D. C. Preston, "Magnetic resonance imaging (mri) of the brain and spine: Basics." https://case.edu/med/neurology/NR/MRI 20Basics.htm, Jul 2016.

[16] "Data." https://www.synapse.org/#!Synapse:syn27046444/wiki/616992.

[17] tbastue (https://stats.stackexchange.com/users/295832/tbastue), "Why do we even need max pooling layers?." Cross Validated. URL:https://stats.stackexchange.com/q/486377 (version: 2020-09-07).

[18] "Source code for monai.networks.nets.segresnet." https://docs.monai.io/en/stable/_modules/monai/networks/ne