Department of Electronic and Telecommunication Engineering

University of Moratuwa

EN3160 - Image Processing and Machine Vision

Brain Tumor Segmentation

Team Achievers

Jayalath K.A.D.M.P	200247P
Kaushalya K.W.K	200298V

Contents

Abstract

This project presents an innovative approach to brain tumor segmentation using advanced image processing and deep learning. Our method achieves considerably higher accuracy in distinguishing tumor regions in brain MRI scans. The results show promise for enhancing medical diagnosis and treatment planning, with potential applications in healthcare. Our code implementation can be found here.

1 Introduction

Brain tumors are a critical concern in medical imaging, often demanding time-consuming manual segmentation. This project presents an innovative method to carry out brain tumor segmentation using advanced image processing and deep learning. Our goal is to improve diagnostic accuracy, reduce the burden on medical professionals, and enhance patient care. In this report, we outline our approach, datasets, and results, while also discussing the significance and potential impact of our implementation in medical imaging and healthcare.

2 Existing alternatives

2.1 ML-based approaches

2.1.1 Support vector machines

SVM is a two-class classification algorithm. One vs rest classifiers are employed to create the multi-class SVM, and the one that separates the test data by a significant margin is chosen. The characteristics extracted from each image are sent into the multi-class SVM trainer, which builds an SVM structure. The structural data is then passed into the SVM classifier, together with the test datum, to provide a categorized image result.

2.1.2 Decision trees

The decision tree algorithm constructs a tree-like structure based on the training data. At each internal node of the tree, the algorithm selects a feature and a corresponding threshold that best splits the data into two subsets with the highest purity. Purity is usually measured using metrics like Gini impurity or information gain. After constructing the tree, pruning techniques can be applied to avoid overfitting. Pruning helps simplify the tree by removing branches that do not significantly contribute to improving classification performance. During testing or inference, the decision tree is used to segment new brain MRI images. The image features are extracted, and the decision tree is traversed. At each node, the extracted feature values are compared to the threshold values stored in the node. The traversal continues along the appropriate branch based on the feature values until a leaf node (a decision) is reached. The decision at the leaf node determines whether the image region corresponds to a tumor or non-tumor region.

2.1.3 K means clustering

K-means aims to partition the data into K clusters, where K is a user-defined parameter representing the number of clusters. We can choose K based on domain knowledge or by

running the algorithm with different K values and evaluating the results. Once K-means converge, it assigns each feature vector to one of the K clusters based on the similarity of the feature vectors. In the context of brain tumor segmentation, the clusters can represent different tissue types or regions, including tumor and non-tumor regions. After the clustering is complete, we can create a segmentation mask by labeling the pixels or regions in the MRI images according to the assigned clusters. For instance, we can label the cluster that corresponds to brain tumors as the "tumor" region, and all other clusters as "non-tumor" regions.

2.2 Advanced deep learning architectures

2.2.1 UNet

U-Net begins by extracting features from the input brain MRI image using a series of convolutional layers. U-Net employs a downsampling or contracting path, which reduces the spatial dimensions of the feature maps. U-Net architecture is the use of skip connections that connect layers at the same resolution in the contracting path to layers in the expanding path. During the upsampling path, the feature maps are combined with the corresponding skip connections from the downsampling path. U-Net ends with a classification layer that produces the final pixel-wise segmentation.

2.2.2 SegNet

The model starts with an encoding or contracting path, consisting of a series of convolutional layers with max-pooling. After the encoding path, SegNet employs a decoding or expanding path that gradually restores the spatial dimensions using deconvolutional layers or upsampling. Similar to U-Net, SegNet also incorporates skip connections. The final layer of the SegNet model is typically a convolutional layer with as many output channels as there are classes.

3 Method

3.1 Data set

The Magnetic Resonance images used for the model training and evaluation are from the Multi-modal Brain tumour Segmentation Challenge (BraTS) 2020 [4]. The BraTS 2020 training dataset contains 369 MR volumes of shape $240 \times 240 \times 155$. MRI is required to evaluate tumor heterogeneity. These MRI sequences are conventionally used for glioma detection: T1 weighted sequence (T1), T1-weighted contrast-enhanced sequence using gadolinium contrast agents (T1CE), T2 weighted sequence (T2), and Fluid attenuated inversion recovery (FLAIR) sequence. From these sequences, four distinct tumor subregions can be identified from MRI as the Enhancing Tumor (ET) which corresponds to the area of relative hyper-intensity in the T1CE with respect to the T1 sequence, Non-Enhancing Tumor (NET), Necrotic Tumor (NCR) which are both hypointense in T1CE when compared to T1, and Peritumoral Edema (ED) which is hyper-intense in FLAIR sequence. These almost homogeneous sub-regions can be clustered together to compose three semantically meaningful tumor classes as, Enhancing Tumor (ET), addition of ET, NET, and NCR represents the Tumor Core (TC) region and addition of ED to TC represents the Whole Tumor(WT). In the dataset, there are masks labeled as [0 1 2 3 4].

But the 3rd mask layer does not contain any data, therefore in the code we have replaced the 3rd mask layer with the 4th mask, the the labels will be [0 1 2 4]. We didn't take the T1 images to the training process because they didn't show any valuable data about the tumour. Therefore we used only T1CE, T2, and FLAIR images. MRI sequences and ground truth map with 4 classes are shown in the below figure.

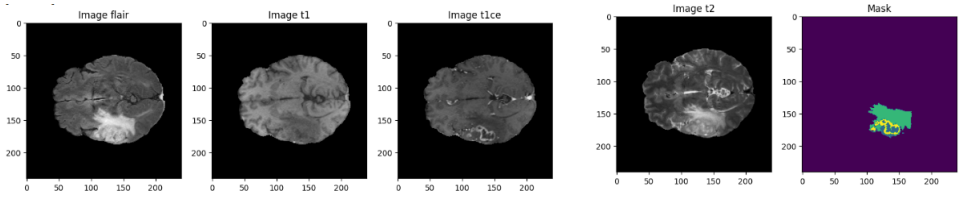

Figure 1: *Visual Analysis of BraTs 2020 Training Data*

3.2 Data pre-processing

In the images lot of space was left empty without any data. Therefore we cropped the images to size 128 x 128 x 128. Then we could obtain a good image containing mostly the brain and the tumour parts. Then we wanted to remove volumes that did not contain useful information in the masks. We removed those volumes that did not have at least 1/100 of useful labels from our training dataset. We used Kaggle to implement our code because of the ease of access to the dataset. Unfortunately, we couldn't load the full 369 MR volumes because of insufficient space(19.5 GB) provided by Kaggle. Therefore we only loaded 130 MR volumes, and we divided the volumes into training and validation at a ratio of 0.75:0.25.

3.3 Network Architecture

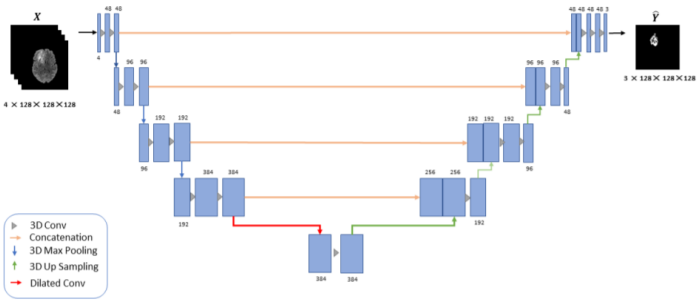

Figure 2: *Segmentation Network - 3D U-net*

The encoder is the initial part of the network and consists of 5 blocks of 3D convolutional layers, followed by non-linear activation functions (ReLU) and 3D max-pooling layers. The encoder progressively reduces the spatial dimensions while increasing the number of feature maps. This hierarchy of feature maps helps the network capture hierarchical features in the input volume.
The decoder is the counterpart of the encoder and is responsible for restoring the spatial dimensions of the feature maps. It consists of 3D transposed convolutional layers that

gradually upsample the feature maps. The decoder blocks also include five 3D convolutional layers and non-linear activation functions. The number of feature maps decreases as we move deeper into the decoder.
The final layer of the 3D U-Net is a 3D convolutional layer with the activation function 'SOFTMAX' because we are applying this to multiclass classification.

3.4 Objective Function

We are defining the loss function, metrics, and optimizer to be used for training a neural network model for segmentation tasks.

3.4.1 Loss Functions

We used two loss functions to calculate the total loss.
Total loss = Dice loss + (1* Focal loss)
Dice Loss is a commonly used loss function for image segmentation tasks. It measures the overlap between predicted and true segmentations. In this case, it is defined with class weights '[wt0, wt1, wt2, wt3]'. These weights can be used to assign different importance to different classes during training.
Focal Loss is another loss function for improving the training of imbalanced datasets. It focuses on hard-to-classify examples by downweighting well-classified examples.
The total loss is defined as the sum of dice loss and focal loss, with the latter having a weight of 1. This combination of loss functions can be useful to balance the trade-off between accuracy and handling class imbalance in segmentation tasks.

3.4.2 Metrics

We used the "MeanIOU" score with a threshold of 0.5 to measure the accuracy of the model. This metric computes the Intersection over Union (IoU) score, which is commonly used for segmentation tasks. It measures the spatial overlap between predicted and true segmentations. The 'threshold=0.5' parameter likely sets a threshold for binarizing the predictions.

3.4.3 Optimizer

The Adam optimizer was used with a learning rate of 0.0001. Adam is a popular optimization algorithm for deep learning models.
These components are essential for configuring and training a deep learning model for 3D image segmentation, and the class weights in the loss function are particularly useful for handling class imbalances that are common in medical image segmentation tasks.

4 Results

The loss value and the MeanIOU accuracy value were calculated at the end of training and validation. The results are shown in the below table.

We trained our model for 100 epochs, and the final accuracy score was around 72%. We only used 75% of 130 MR volumes(around 97 volumes) for training the model because of insufficient data storage. If we could access more storage space and trained our model for more volumes we could have achieved a higher accuracy level.

	Loss	MeanIOU
Training	0.7837	-
Validation	0.7840	0.7237

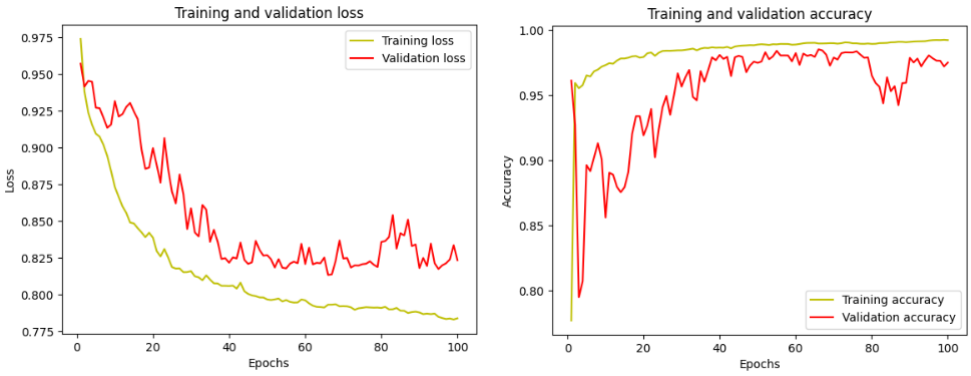

Figure 3: *Loss and Accuracy graphs of Training and Validation*

Most of the models we referred to, used only the Dice loss as the loss function. We combined both Dice loss and Focal loss to get a more suitable loss function for this application.

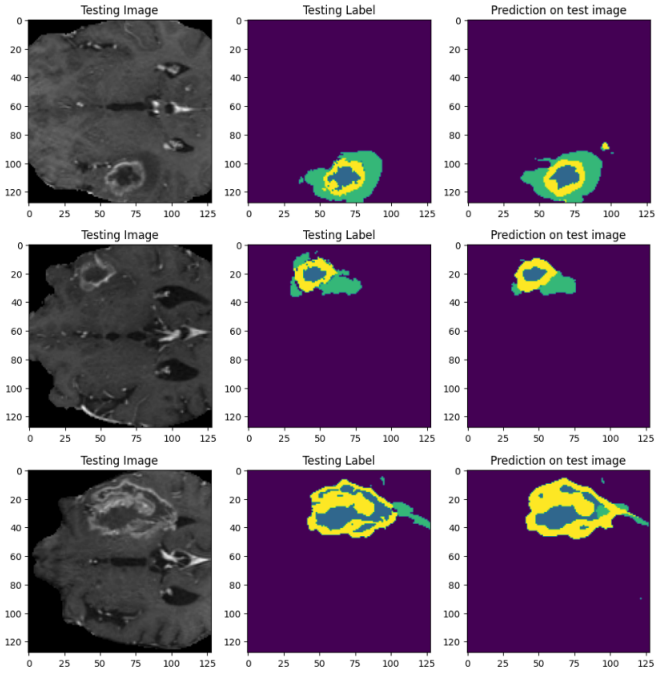

Figure 4: *Results*

5 Discussion

Here we would like to discuss some of the challenges we faced and possible directions for future work and research.

- The main problem we faced was insufficient storage space in Kaggle when loading the dataset. We would suggest Colab and upload the dataset in a cloud resource.

- Also, we can improve the model using other upgraded 3D models such as UNET++
- We can also create loss functions for each label separately and train the model more accurately.
- Recently using Generative Adversarial Networks(GANs) has been a major breakthrough in the image generation task. We can apply those concepts to further improve our implementation.

6 Conclusion

In conclusion, this project has demonstrated the efficacy of using advanced machine vision techniques, including deep learning models, for brain tumor segmentation in MRI images. Our model has shown accurate results and the potential to improve the diagnostic process. While challenges remain, such as data storage and high-performance GPU access, our work provides a strong foundation for future research and enhancements in this critical field of medical imaging.

References

[1] BraTS 2020 Dataset -https://www.kaggle.com/datasets/awsaf49/brats20-dataset-training-validation.

[2] SVM - https://www.sciencedirect.com/science/article/pii/S2665917422000745#sec2.

[3] Decision tree - https://www.sciencedirect.com/science/article/abs/pii/B9780128222713000128.

[4] K means clustering - https://www.sciencedirect.com/science/article/abs/pii/B9780323983709000202.

[5] Brain tumor segmentation using MRI multi-modalities brain images - https://www.nature.com/articles/s41598-021-90428-8.

[6] Similar Model - https://github.com/himashi92/vizviva_brats_2021.

[7] Reciprocal Adversarial Learning for Brain Tumor Segmentation - https://arxiv.org/pdf/2201.03777.pdf.

[8] Brain Tumor Segmentation using the UNET Architecture in TensorFlow - https://www.youtube.com/watch?v=lstBIXVUoSM