EXPLORING FEW-SHOT LEARNING FOR BRAIN TUMOR SEGMENTATION

A PROJECT REPORT

Submitted by

Sannapaneni Kundan Sai Chowdary (195002103)
Pavan Kumar Malasani (195002080)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION TECHNOLOGY

Department of Information Technology

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

MAY 2023

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project titled "**EXPLORING FEW-SHOT LEARNING FOR BRAIN TUMOR SEGMENTATION**" is the bonafide work of "**Sannapaneni Kundan Sai Chowdary** (195002103), and **Pavan Kumar Malasani** (195002080)", who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. C. ARAVINDAN
PROFESSOR AND HEAD
Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
Kalavakkam – 603110

Dr. C. ARAVINDAN
SUPERVISOR
Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
Kalavakkam – 603110

Submitted for project viva-voce examination held on……

EXTERNAL EXAMINER **INTERNAL EXAMINER**

ABSTRACT

Brain tumor segmentation is an important problem in the field of image segmentation, with practical applications in both academic oncology and surgery. Accurate segmentation of tumor subregions can greatly benefit these fields, and the BraTS (Brain Tumor Segmentation) dataset is a valuable resource for research in this area. The BraTS dataset includes detailed manually labeled 3D MRI scans of brain tumors, which can be used to train and test deep learning models for accurate segmentation.

In our study on brain tumor segmentation, we presented a modified deep learning model called RU-Net, which accurately segments brain tumors within certain computational constraints. However, the limited availability of labeled data posed a challenge. To overcome the challenge of limited labeled data, we employed a teacher-student model, where the student model was trained using the labeled data, and selecting only the most informative parameters that contribute significantly to the model's output. By using the student model to generate pseudo-labels for additional data, we were able to further improve the model's accuracy.

Moreover, to optimize our model's performance in the future, we plan to incorporate pre-trained weights from an autoencoder. By using a pre-trained autoencoder, we can initialize our deep learning model

with weights that have already learned useful features from the dataset, thereby improving the efficiency of training and potentially improving the accuracy of the model. Our study demonstrates the potential of using machine learning techniques to improve medical diagnosis and treatment. The RU-Net model achieved accurate segmentation results, with the whole tumor achieving a dice score of 0.7234 and the necrotic, enhancing, and edema subregions achieving dice scores of 0.7172, 0.7863, and 0.8557, respectively.

The RU-Net model is based on the U-Net architecture, which is a popular and effective deep learning architecture for image segmentation tasks. In addition, we added RES (Residual Extended Skip) blocks in the concatenation pathways to control informational degradation, which improved the dice coefficients of the tumor subregions' segmentations.

Our study demonstrates that the RU-Net model can accurately segment brain tumors within certain computational constraints, making it a valuable tool for medical professionals in the fields of academic oncology and surgery. Our study also highlights the importance of deep learning and image segmentation in medical research, and the potential benefits of using machine learning techniques to improve medical diagnosis and treatment.

ACKNOWLEDGEMENT

We would like to express our sincere thanks to all those who provided us with the possibility to complete this project. We are highly indebted to Sri Sivasubramaniya Nadar College of Engineering and our Principal, **Dr. V.E. Annamalai**, for their guidance and support by providing the facilities to carry out this project.

We would like to express our sincere gratitude to **Dr. C. Aravindan,** who has served as both our Head of the Department, and Guide throughout the project. As a Professor in the Department of Information Technology, his valuable insights and guidance have been instrumental in the success of our project. His expert mentorship, suggestions, and feedback have been priceless, and we are grateful for his constant support and encouragement.

We are deeply indebted by the suggestions and guidance provided by our project panel members of our department, **Dr. N. Bhalaji**, Associate Professor and, **Dr. I. Joe Louis Paul**, Associate Professor, and **Dr. V. Thanikachalam,** Associate Professor, for their help and feedback during the various stages of our project.

We are thankful for the support and encouragement provided by the Department of Information Technology during the course of our project. Finally, with great enthusiasm, we express our thanks to all our department faculty and technical staff, friends and parents for their sustained support and interest in the completion of our project.

Sannapaneni Kundan Sai Chowdary

Pavan Kumar Malasani

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

Our project aims to create a modified model based on Active learning technique that can perform precise automatic brain tumor segmentation. To achieve this, we will utilize multi-institutional pre-operative multimodal MRI scans of patients diagnosed with glioma. The model must accurately identify the tumor from among the intricate brain tissues and segment it into different subregions of clinical significance, as we will describe in the following sections.

1.2 MOTIVATION

Magnetic Resonance Imaging (MRI) and resection surgery are the standard diagnostic and treatment approaches for brain tumors. Precisely marking the tumor region is a top priority for neurosurgeons as it serves as a critical guide during surgical procedures. However, manual segmentation of tumors is a labor-intensive task that is susceptible to human errors, especially given the challenges of identifying tumors using the naked eye alone.

Another major obstacle in practical application is that these models tend to take up a large amount of space and perform segmentations slowly, which limits their usability. Consequently, many medical institutions still rely on manual segmentation. To address these challenges, our project aims to train our model for as few epochs as possible while still achieving accurate segmentation results.

Thus, our ultimate goal is to develop a model that can perform accurate automatic brain tumor segmentation, which can aid neurosurgeons and other medical professionals in their work. By tackling this issue, we hope to provide a more efficient and effective alternative to manual segmentation.

1.3 MEDICAL IMAGE SEGMENTATION

Image segmentation is a technique used to partition a digital image into smaller subgroups, called image segments. This process reduces the complexity of an image and facilitates its analysis or processing. The primary aim of segmentation is to simplify or transform the image's representation into a more meaningful format that is easier to analyze. Segmentation is often utilized to locate objects and boundaries, such as lines and curves, within an image. Specifically, image segmentation involves assigning a label to each pixel in an image based on shared characteristics, so that pixels with the same label have similar properties. This label assignment process helps to segment the image into distinct regions and enable further analysis of the segmented image.

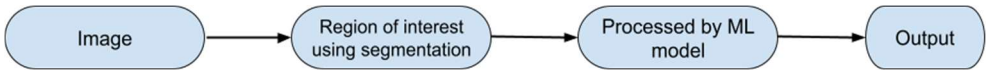

Fig. 1.1. A simple outline of the image segmentation process.

The process of medical image segmentation entails identifying regions of interest (ROIs) within 3D image data, such as those obtained from Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scans. The primary objective of segmenting such data is to pinpoint anatomical regions that are necessary for a specific study.

1.4 BRAIN TUMOR SEGMENTATION

A brain tumor is an abnormal growth of cells in the brain, which is a highly intricate organ responsible for various nervous system functions. Brain tumors can emerge in any part of the brain or skull, including the protective lining, skull base, brainstem, sinuses, nasal cavity, and other regions. With more than 120 types of brain tumors originating from different tissues, many of them are considered the most severe types of tumors globally. Glioma, which is the most common primary brain tumor, is caused by the abnormal growth of glial cells in the spinal cord and brain. It is characterized by varying grades of malignancy and histology, with patients diagnosed with glioblastoma surviving less than 14 months on average.

Magnetic Resonance Imaging (MRI) is a non-invasive technique that produces a wide range of tissue contrasts in each imaging modality and is commonly used by medical professionals to diagnose brain tumors. However, the manual segmentation and analysis of structural MRI images of brain tumors are arduous and time-consuming tasks that require trained neuroradiologists. This process is crucial for treatment planning and determining the tumor's extent and location. An accurate segmentation provides a plan for neurosurgeons to remove as much of the tumor as possible while minimizing damage to healthy brain tissue.

An automatic and robust brain tumor segmentation method can have a substantial impact on the diagnosis and treatment of brain tumors. It can enable the timely treatment of neurological disorders such as Alzheimer's disease (AD), schizophrenia, and dementia. In addition, an accurate segmentation can provide valuable information for monitoring tumor growth and response to therapy. Thus, the development of an automatic brain tumor segmentation method can contribute to improving patient outcomes and enhancing the overall quality of healthcare.

Brain tumor segmentation is the task of segmenting tumors from other brain artefacts in MRI image of the brain. To segment the gliomas in pre-operative MRI scans for aid in surgical procedures the following sub-regions of the tumor are considered:

1) **the "enhancing tumor" (ET)**,

2) the "tumor core" (TC)

3) the “edema or tumoral invasion” (ED)

3) **the "whole tumor" (WT)**.

The four distinct tumoral subregions defined from MRI are:

1) **the “enhancing tumor” (ET)**: It is the subregion of the tumor that is able to increase vascularization or to be responsible of neoangyogenesis (formation of new blood vessels), inflamatory or tumoral. To put it simply, it is the actively spreading cancerous region.

2) **the “non enhancing tumor” (NET)**: It is the tumor subregion that lacks vascularization or neoangyogenesis. It is rarely malignant.

3) **the “necrotic tumor” (NCR)**: Tumor necrosis is the foci of necrotic cell death, occurs in advanced solid tumors and is often associated with poor prognosis of cancer patients.

4) **the “peritumoral edema” (ED)**: Peritumoral edema is the swelling caused by excess fluid trapped in the tissues surrounding the tumor core. It is a characteristic feature of malignant glioma related to the extent of neovascularization.

ET is the first cluster. The addition of ET, NET and NCR represents the “tumor core” (TC) region. The addition of ED to TC represents the “whole tumor” (WT). [5] The tumoral clusters and subregions discussed here are visualized in the next page in Fig. 1.2. and Fig. 1.3. by utilizing the samples available through the BraTS dataset.

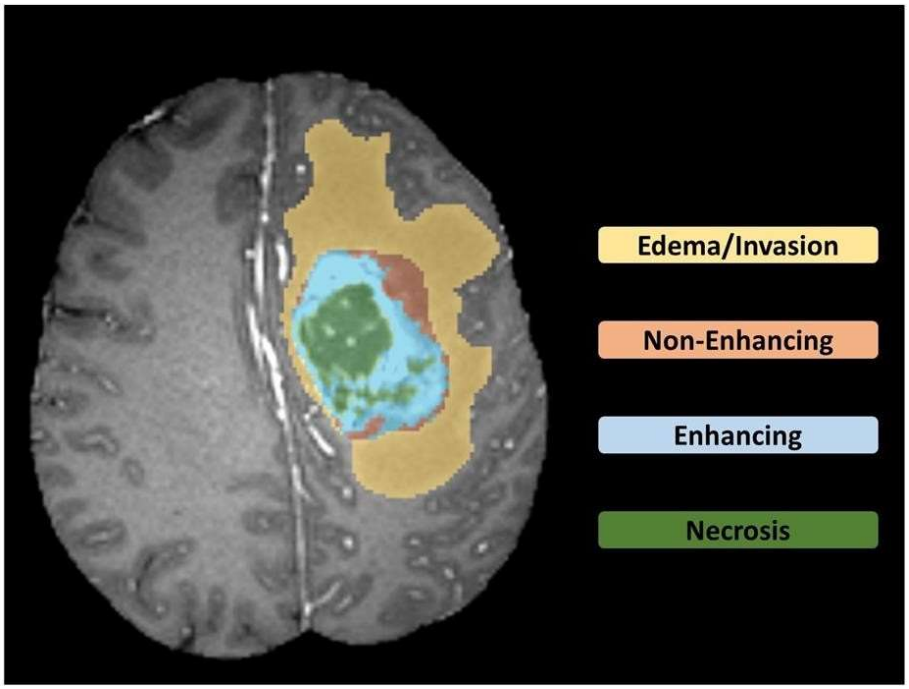

Fig. 1.2. An illustration of the different tumoral subregions that we work with and are available to us through the 2020 BraTS Dataset. [5]

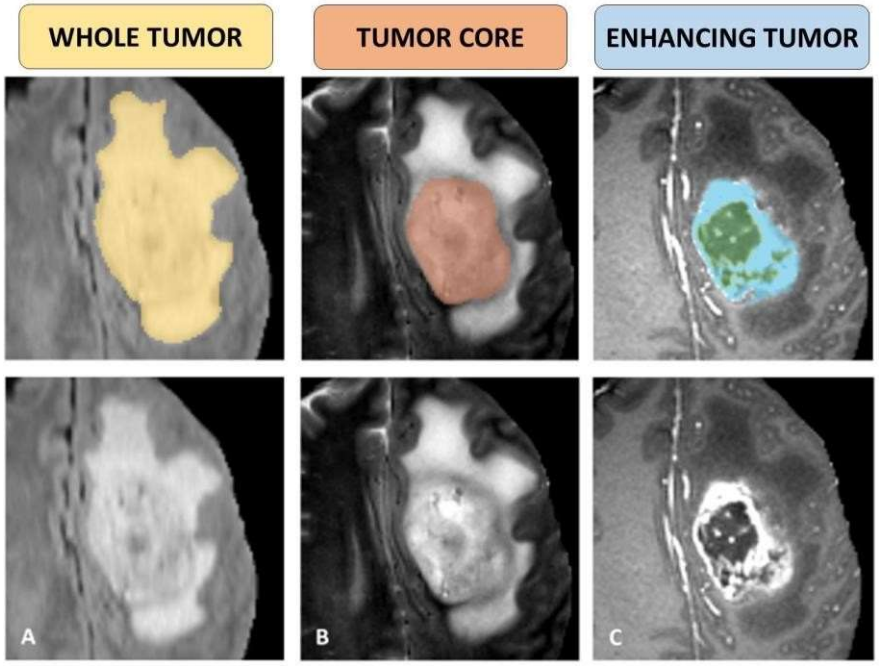

Fig. 1.3. The three tumoral clusters we identify and utilize for the segmentation task namely, WT, TC and ET. [5]

1.5 BRATS DATASET

The Center for Biomedical Image Computing and Analytics (CBICA) initiated BraTS, which encourages participants to develop competitive solutions for brain tumor segmentation tasks. The BraTS dataset consists of multimodal 3D brain MRIs and ground truth brain tumor segmentations annotated by physicians. It includes 4 MRI modalities per case (T1, T1ce, T2, and FLAIR). For this project, the BraTS Dataset of 2020, which is publicly available, was used.

An MRI sequence is a set of radiofrequency pulses and gradients that produce a series of images with a distinct appearance, forming the basis for classifying different MRI modalities. The Repetition Time (TR) is the interval between successive pulse sequences applied to the same slice, while the Time to Echo (TE) is the time between the delivery of the RF pulse and the receipt of the echo signal. Differences in TR and TE result in various modalities.

The BraTS dataset includes multimodal 3D brain MRIs that are available as NIfTI files (.nii.gz) and describe the following modalities:

a) **T1**: T1 (longitudinal relaxation time) is a measure of the time taken for excited protons to return to equilibrium with the external magnetic field. T1-weighted images are produced using short TE and TR times, and the contrast and brightness of the image are predominantly determined by T1 properties of tissue.

b**) Contrast enhanced T1 (T1ce):** T1ce is a T1 sequence coupled with injections of gadolinium (Gd) as a contrast agent. Gd is a non-toxic paramagnetic contrast enhancement agent that changes signal intensities by shortening T1 when injected during the scan.T1ce images are especially useful in looking at vascular structures and breakdown in the blood-brain barrier, such as tumors, abscesses, and inflammation.

c) **T2:** T2 (transverse relaxation time) is a measure of the time taken for excited protons to lose phase coherence among the nuclei spinning perpendicular to the main field. T2-weighted images are produced using longer TE and TR times, and the contrast and brightness are predominantly determined by the T2 properties of tissue.

d) **Fluid Attenuated Inversion Recovery (FLAIR):** FLAIR is similar to a T2-weighted image, but with very long TE and TR times. By doing so, normal cerebrospinal fluid (CSF) is attenuated and made dark, while abnormalities remain bright. This sequence is very sensitive to pathology and makes the differentiation between CSF and an abnormality, such as parenchymal edema, much easier.

The ET refers to areas of the brain that appear brighter in T1ce than in T1, and also brighter than healthy white matter in T1Gd. The TC refers to the main mass of the tumor that is typically removed during surgery, and includes both the ET and the necrotic parts of the tumor, which appear darker in T1-Gd than in T1. The WT is a more comprehensive measure of the disease, as it includes the TC and the tissue around the tumor that has been invaded or swollen, which is shown as a brighter signal in FLAIR. [5]

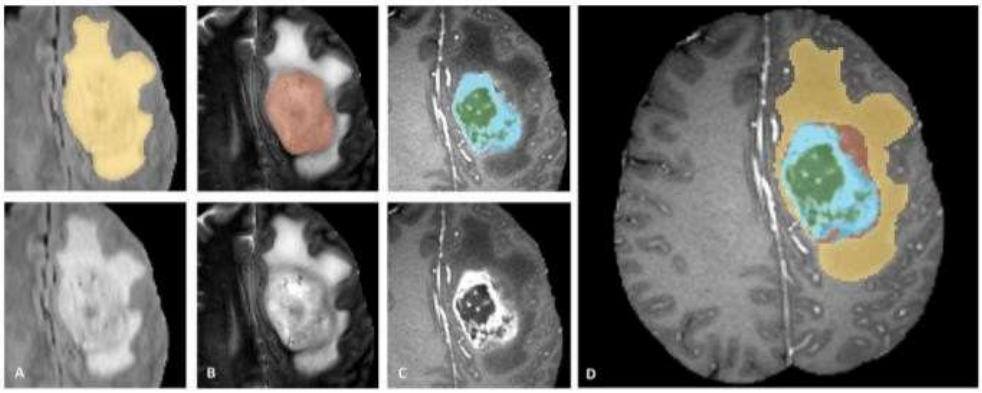

Fig. 1.4. A visualization of the BraTS Dataset tumor subregion ground truth. [5]

The BraTS dataset of 2020 includes MRI scans of brain tumors that were acquired using different clinical protocols and various scanners from over 19 institutions. To ensure consistency in the annotations, all tumors in the MRIs were manually segmented by one to four raters following the same annotation protocol, and their annotations were approved by experienced neuro-radiologists. The annotations include the GD-enhancing tumor (ET - label 4), peritumoral edema (ED - label 2), necrotic and non-enhancing tumor core (NCR/NET - label 1), and the whole tumor (WT – label 3), as described in the BraTS 2012-2013 TMI paper and the latest BraTS summarizing paper.

The provided data in the BraTS dataset are pre-processed, meaning they are co-registered to the same anatomical template, interpolated to the same resolution (1 mm^3), and skull-stripped. All slices of volumes are converted to hdf5 format to save memory. Metadata includes volume number, slice number, and target of that slice, providing essential information for researchers to access the data easily.

The manual segmentations available in the BraTS dataset serve as ground truth and are crucial for assessing the effectiveness of segmentation algorithms. In Fig. 1.4, image patches with annotated tumor structures in different modalities are shown, along with the final labels for the entire dataset. The image patches depict the whole tumor visible in FLAIR (A), the tumor core visible in T2 (B), the enhancing tumor structures visible in T1c (blue), and the surrounding cystic/necrotic components of the core (green) (C). The segmentations are combined to generate the final labels of the tumor structures (D): edema (yellow), non-enhancing solid core (red), necrotic/cystic core (green), and enhancing core (blue). The ground truth provided in the BraTS dataset is an excellent benchmark for evaluating the performance of segmentation models.

CHAPTER 2

LITERATURE SURVEY

In this chapter, we list out and discuss the previous works published in the field of Brain Tumor Segmentation. We will take a look at the architectures used to address the problem statement and arrive at a conclusion on how our architecture should be.

1) The Multimodal Brain Tumor Image Segmentation Benchmark (BraTS) (2015) Menze B. H., Jakab A., Bauer S., Kalpathy-Cramer J., Farahani K., Kirby J., et al.

This paper details the set-up and results of the Multimodal Brain Tumor Image Segmentation Benchmark (BraTS). It serves as the perfect launching pad to initiate our research into the literature of the field of Brain Tumor Segmentation. The paper introduces us to first efforts in this field by discussing the problem of segmenting gliomas.

Gliomas are the most commonly occurring primary brain tumors in adults, and are believed to originate from glial cells while infiltrating the surrounding tissues. Despite significant advances in glioma research, patient diagnosis remains challenging. The current clinical routine and many clinical studies rely on the qualitative assessment of images, based solely on the visual appearance of hyper-intense tissue in contrast-enhanced T1-weighted MRI. Similarly, rudimentary quantitative measures, such as the largest diameter visible from axial images of the lesion, are also used in diagnostic evaluations.

These approaches can be limiting and may not provide a comprehensive assessment of the extent and severity of the tumor. Therefore, there is a critical need for more sophisticated and accurate methods of diagnosis, particularly for gliomas. Such methods can lead to better patient outcomes by allowing for earlier and more precise diagnoses, and ultimately guiding more effective treatment strategies.

It is evident that replacing the current rudimentary assessments with more accurate and consistent measurements of the relevant tumor substructures would be highly beneficial for the diagnosis, treatment planning, and follow-up of individual patients with brain tumors. The potential value of image processing routines that can automatically analyze brain tumor scans is enormous, as they can provide highly accurate and reproducible measurements of tumor substructures. However, the main challenge is that many of the current state-of-the-art algorithms for tumor segmentation are not developed specifically for gliomas but are based on techniques developed for other structures or pathologies. For instance, automated white matter lesion segmentation algorithms have reached considerable accuracy, but they are not specifically designed for gliomas. In this context, the present paper represents a significant breakthrough by introducing and documenting novel glioma-specific tumor segmentation techniques that can be used for improving the diagnosis, treatment planning, and follow-up of glioma patients. The paper provides a detailed discussion of these novel techniques and their potential implications for glioma research and practices.

In this paper, the authors demonstrate the performance of twenty advanced tumor segmentation algorithms on a dataset consisting of 65 multi-contrast MR low- and high-grade glioma scans. These scans were manually annotated and were also generated using tumor image simulation software to provide a robust evaluation. The paper presents innovative techniques such as the use of voxels from multiple MRI modalities, including FLAIR, T1, T1C, T2, and neighboring voxels, to create a feature vector that is inputted into the model for training. The model then produces a predicted label for each feature vector. The authors also explore various post-processing techniques, such as filtering out small objects and applying dilation and erosion operations on each segmented class. The study showcases the potential of these advanced techniques for improving glioma segmentation accuracy and highlights the importance of optimizing these methods for specific pathologies.

In summary, the paper suggests that different algorithms demonstrate varying levels of success in segmenting different tumor sub-regions. The study found that the highest dice scores achieved were 59.5% for edema and 65.6% for the whole tumor. It was noted that no single algorithm was able to perform consistently well in all sub-regions simultaneously, which highlights the challenges and trade-offs that are inherent in tumor classification. The study also identified certain limitations in the algorithms that were used, including their inability to support spatial features, neighborhood-based textural features, and the use of atlas-based priors, which have been shown to improve segmentation accuracy. However, these limitations were overcome through the introduction of deep neural networks, and this is discussed in detail in the subsequent paper.

2) Brain Tumor Segmentation with Deep Neural Networks (2017)
Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M. and Larochelle, H.

The paper presents a novel approach for brain tumor segmentation using Deep Neural Networks (DNNs) specifically designed for glioblastomas, which can be low or high grade, in MR images. The proposed network employs Convolutional Neural Networks (CNNs), which are a type of DNN, for accurate segmentation. While CNNs have been previously applied to segmentation tasks, most of the work has focused on non-medical tasks, and the architectures used are not well-suited to medical imagery or brain tumor segmentation.

The authors present a CNN architecture that utilizes the latest advancements in CNN design and training techniques, such as Maxout hidden units and Dropout regularization.

They evaluated their approach on the fully-annotated MICCAI brain tumor segmentation (BRATS) challenge 2013 dataset, using the well-defined training and testing splits. The results demonstrate the effectiveness of their approach in accurately segmenting brain tumors and outperforming other state-of-the-art methods.

Their contributions in this work are four-fold:

1. The authors developed a fully automatic brain tumor segmentation method using Deep Neural Networks (DNNs) that achieved the second-highest score on the 2013 BRATS challenge leaderboard.
2. The proposed method is significantly faster than most state-of-the-art methods, taking between 25 seconds and 3 minutes to segment a brain.
3. The CNN architecture used in the method incorporates a novel two-pathway design that considers both local details and larger context. The authors also implemented a two-phase training procedure that is effective in handling imbalanced label distributions.
4. The authors introduced a novel cascaded architecture that serves as an efficient and clean alternative to other popular structured output methods.

The figure demonstrates the input patch being processed through two convolutional paths, referred to as the local and global paths. The local and global feature-maps are depicted in yellow and orange, respectively. The dashed lines in the figure indicate the convolutional layers utilized to generate these feature-maps. The TwoPathCNN model is encapsulated within the green box and is utilized in subsequent architectures.

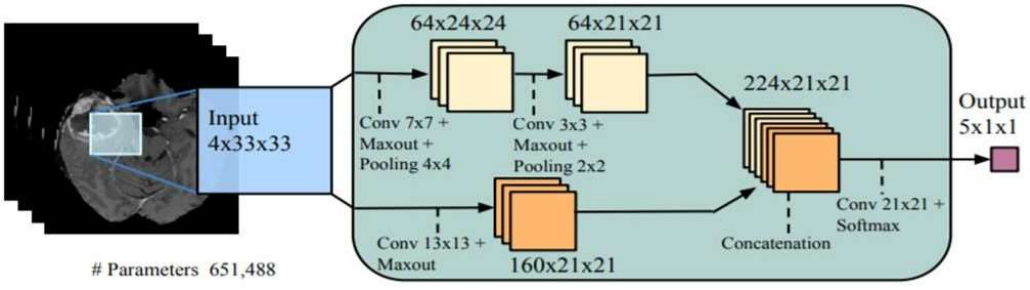

Fig. 2.1. Two-pathway CNN architecture (TwoPathCNN).

While the TwoPathCNN architecture proposed in the previous paper is impressive, it has a limitation. Many machine learning methods, including this architecture, perform pixel classification without considering the local dependencies of labels. This means that segmentation labels are conditionally independent given the input image, which can be problematic.

To address this issue, one can model label dependencies by using the pixel-wise probability estimates of an initial CNN as additional input to certain layers of a second DNN, creating a cascaded architecture. This approach is faster than implementing a conditional random field (CRF), a popular structured output method for modeling label dependencies. The reason is that convolutions are efficient operations that can be applied to the pixel-wise probability estimates.

However, as will be discussed in the next paper, there are advantages to using CRFs as a structured output method. These include the ability to capture higher-order dependencies between labels, which can improve segmentation accuracy. Therefore, the choice of structured output method depends on the specific task and requirements of the application.

3) Efficient Multi-Scale 3D CNN with fully connected CRF for Accurate Brain Lesion Segmentation (2017) Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D. and Glocker, B.

The paper titled "Efficient Multi-Scale 3D CNN with fully connected CRF for Accurate Brain Lesion Segmentation" proposes an improved approach for brain lesion segmentation using a combination of convolutional neural networks (CNNs) and conditional random fields (CRFs). The proposed method outperforms previous methods and achieves high accuracy in brain lesion segmentation. The authors propose a multi-scale 3D CNN architecture that effectively captures both local and global information. They also incorporate a fully connected CRF to model the dependencies between adjacent voxels and improve the segmentation accuracy. The proposed method achieves state-of-the-art results on the publicly available MICCAI BRATS 2015 and 2013 datasets.

The authors provide a detailed description of their proposed method and present experimental results on the BRATS datasets. They show that their method outperforms several state-of-the-art methods in terms of accuracy, sensitivity, and specificity. The proposed method also has a faster runtime compared to other methods. The paper provides visual examples of the segmentation results, demonstrating the accuracy of the proposed method. The authors also provide a comprehensive analysis of the proposed method's strengths and limitations, and they suggest future research directions to improve brain lesion segmentation further.

They analyzed the development of deeper, thus more discriminative 3D CNNs. In order to incorporate both local and larger contextual

information, we employ a dual pathway architecture that processes the input images at multiple scales simultaneously. The use of a 3D fully connected Conditional Random Field for post processing effectively removes false positives giving more dice coefficient. This architecture is shown below in Fig. 2.2.

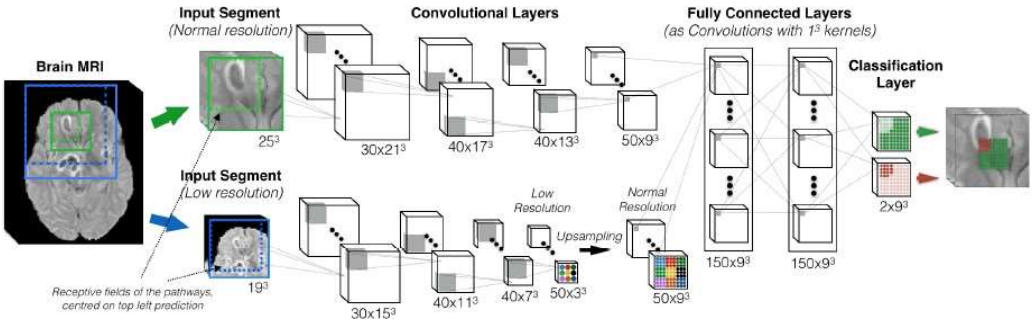

Fig. 2.2. Multi-scale 3D CNN with two convolutional pathways.

The pipeline, trained for 400 epochs, is extensively evaluated on three challenging tasks of lesion segmentation in multi-channel MRI patient data with traumatic brain injuries, brain tumors, and ischemic stroke. The authors improved on the state-of-the art for all three applications, with top ranking performance on the public benchmarks BRATS 2015 and ISLES 2015. This method is computationally efficient, which allowed its adoption in a variety of research and clinical settings. The dice scores obtained are tabulated below in Table 2.1.

Table. 2.1. Average dice coefficient values of the 3D CNN system on BRATS 2015. [4]

	DSC		
	WHOLE	**CORE**	**ENH.**
DeepMedic	83.6	67.4	62.9
DeepMedic+CRF	84.7	67.0	62.9
Ensemble	84.5	66.7	63.3
Ensemble+CRF	84.9	66.7	63.4

DeepMedic here refers to the unique CNN architecture developed by the authors and DeepMedic+CRF is the model coupled to the CRF at the output. In addition to a singular model, they've also trained an ensemble of 10 models for which the dice score is also mentioned. Despite the dice score being very remarkable there was yet another innovation which changed the paradigm in brain tumor segmentation. That was the U-Net.

4) U-Net: Convolutional Networks for Biomedical Image Segmentation (2015) Ronneberger, O., Fischer, P. and Brox, T.

The U-Net was invented to deal with biomedical images where the goal is not only to classify whether there is an infection or not but also to identify the area of infection. It is a convolutional neural network based architecture that expanded with few changes in the existing CNN

architecture leading to a more “fully convolutional network”. The authors modified and extended this architecture such that it works with very few training images and yields more precise segmentations. In total the network has 23 convolutional layers. The main idea is to supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. These layers increase the resolution of the output. Consequently, the architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization.

To achieve precise localization, the paper proposes combining high resolution features from the contracting path with the upsampled output, followed by a convolution layer that can learn to assemble a more accurate output using this information. A key modification in their architecture is the use of a large number of feature channels in the upsampling part, which allows the network to propagate context information to higher resolution layers. This results in a u-shaped architecture that is more or less symmetric to the contracting path.

The neural network utilized in this study does not employ any fully connected layers, and instead solely relies on the valid portion of each convolution. As a result, the segmentation map only includes pixels that have complete context in the input image. This approach enables the smooth segmentation of images of any size by utilizing an overlap-tile technique. To account for the pixels in the border region of the image, the network extrapolates the missing context by reflecting the input image. The application of this tiling method is crucial when working with larger images, as the resolution would otherwise be restricted by GPU memory.

When it comes to the task of segmenting brain tumors, there is a scarcity of available training data. To address this issue, a technique called data augmentation is employed, where elastic deformations are applied to the existing training images. By doing so, the neural network can learn to be invariant to these deformations without needing to observe them in the annotated image dataset. This is particularly crucial in biomedical segmentation, as tissue deformation is a common variation, and realistic deformations can be efficiently simulated.

Another challenge in many cell segmentation tasks involves accurately separating touching objects of the same class. To overcome this, a weighted loss approach is proposed. In this approach, background labels that separate touching cells are assigned a higher weight in the loss function. The resulting network can be applied to various biomedical segmentation problems. The figure below (Fig. 2.3) illustrates the schematic of the conventional U-Net architecture. Each blue box represents a multi-channel feature map, with the number of channels indicated at the top of the box. The x-y size is provided at the lower left edge of the box. White boxes indicate copied feature maps, and the arrows represent different operations.

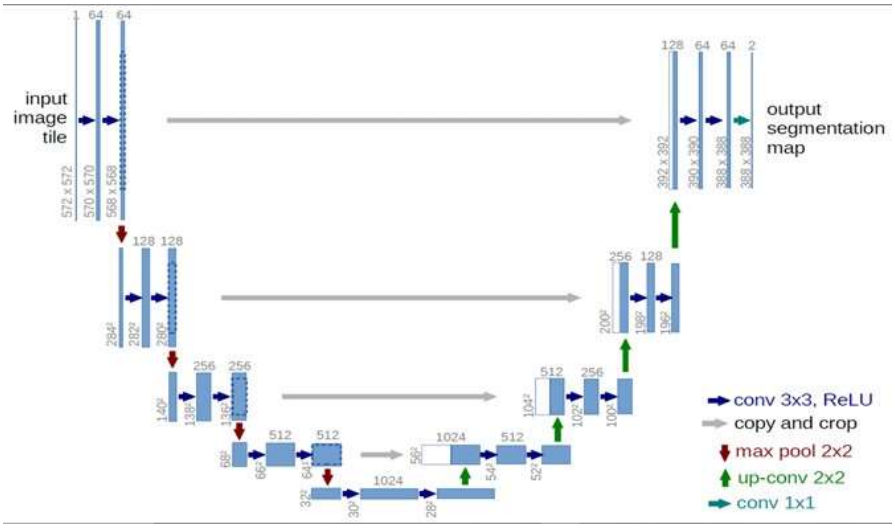

Fig. 2.3. U-net architecture.

The U-Net architecture quickly gained popularity in the field of brain tumor segmentation once its potential was realized. It has since become one of the most widely used architectures for this purpose. The next paper focuses on a specific application of U-Net in brain tumor segmentation and provides a detailed analysis of its performance and results.

5) 3D U-Net Based Brain Tumor Segmentation and Survival Days Prediction (2019) Wang, F., Jiang, R., Zheng, L., Meng, C. and Biswal, B.

In this work, focuses on addressing two tasks within the BraTS 2019 challenge: tumor segmentation and survival days prediction. The researchers employ the U-Net architecture to tackle these tasks. They propose a deep learning model based on the 3D U-Net, which is trained using normalization and patching strategies specifically designed for brain tumor segmentation.

Instead of normalizing voxel values across the entire image, the normalization and scaling process is applied only to the brain area, excluding the black background. The training process involves two phases, each utilizing different patching strategies. One phase takes into account the black background while the other does not, both using a patch that covers the center field.

The tumor images that are generated through the tumor segmentation task are subsequently utilized for predicting the overall survival (OS) days.

To estimate the OS days, a fully connected neural network model is employed, consisting of two hidden layers, each with 64 filters. The accuracy of the model is determined by implementing a three-category classification system using this architecture. Specifically, survival days less than 300 are classified as short-survival, days ranging from 300 to 450 are classified as mid-survival, and days exceeding 450 are classified as long-survival. The accuracy value is calculated based on this classification scheme.

Table. 2.2. shows the mean values of dice coefficient for various tumor

subregions obtained by training the network with two patching strategies each for 100 epochs (the second phase use the first phase saved model as pre-trained), thus effectively training for 200 epochs in total.

Table. 2.2. Dice coefficient obtained with U-Net

DATASET	**DICE**		
	ET	**TC**	**WT**
Training	0.830	0.888	0.916
Validation	0.737	0.807	0.894
Final Test	0.778	0.798	0.852

Summary

Thus, from the literature survey we've traced the evolution of ideas in solving the hard problem of brain tumor segmentation. We have covered the preliminary methods that utilized mere algorithms to segment brain tumors. [5] Then, we moved on to the use of deep neural networks like CNN and saw how modifying them will result in better dice scores. [4] Finally, we had a thorough look at the U-Net which has now become the standard for segmentation tasks. [9] From the progress of CNN based models that came before U-Net, we learn the importance of architectural modification and we will apply the same principle to U-Net. While many research works are currently being directed towards this direction, we will look into the available theory of some new structural advancements that can help us come up with an efficient modified U-Net architecture.

CHAPTER 3

THE RU-NET ARCHITECTURE

In this chapter we will look into the latest available structures that are introduced in theory that could help us with creating a suitable modified U-Net for our purpose. We will then proceed to implement this modified U-Net and discuss the architecture we have created.

3.1 RESIDUAL EXTENDED SKIP (RES) BLOCK

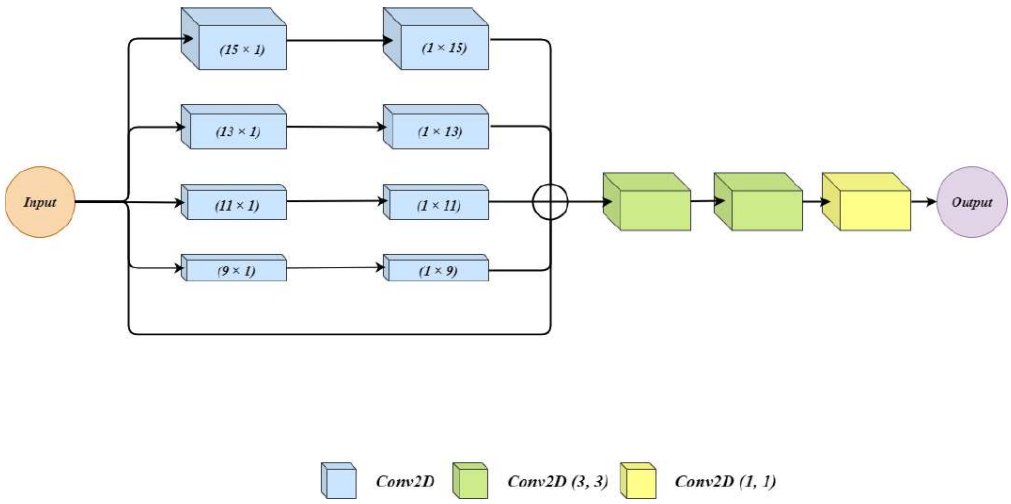

Fig. 3.1. Schematic visualization of a simple RES (Residual Extended Skip) block. [7]

Fig. 3.1. shows the architecture of the residual extended skip (RES) block that we will be including in our architecture. It was first theoretically conceived by Mobeen Ur Rehman et al (2020) to combat various limitations in the U-Net architecture. The main ones are the loss of

valuable information along the concatenation paths between the contracting and expanding pathways in the U-Net. This causes a lot of damage to the quality of segmentation and if solved it could greatly increase the dice score. The other issue is something that's intrinsic to the task of brain tumor segmentation itself, that the dataset is highly imbalanced. This is due to the fact that each and every tumor is unique and is completely unrelated to any other tumor. Consequently, due to the great variations in size, shape and location of the tumors it is difficult for models to learn to segment them.

The RES block is an effective solution for addressing the issues mentioned. The U-Net architecture implements the RES block at each of its concatenation paths. The input to the RES block is passed through five parallel connections, where the first four connections consist of two convolution layers each. The first convolution layer in each connection uses an N x 1 filter size, while the second convolution layer uses a 1 x N filter size. By using cascaded convolution layers, the number of parameters generated is reduced, which is beneficial for the overall architecture. The impact of cascaded convolution layers with a lower number of parameters is similar to that of a single convolution layer with a higher number of parameters. Finally, the last connection is a skip connection, where the input is forwarded as it is. [7]

The five parallel connections in a RES block are summed to obtain a single output, which is then subjected to three consecutive convolution layers. These layers use filter sizes of 3 x 3, 3 x 3, and 1 x 1. By placing RES blocks in the concatenating paths of the U-Net, the low-level features are combined with middle-level features, preventing information degradation in the pathways.

The RES block is also effective in addressing the problem of dataset imbalance caused by high size variations in cancer regions. The residual extended skip performs contextual aggregation on multiple scales, which makes the architecture scale-invariant. This approach increases the valid receptive field, which is not always dominant in previous methods, leading to more accurate segmentation results. [7]

3.2 WIDE CONTEXT (WC) BLOCK

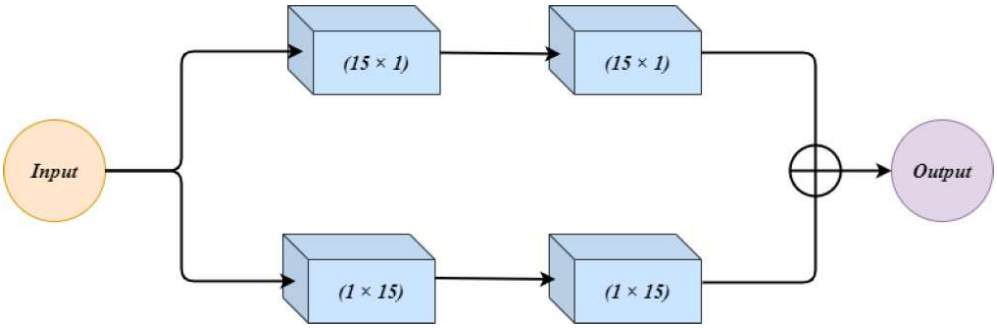

Fig. 3.2. Schematic visualization of the WC (Wide Context) block. [7]

Fig. 3.2. shows the architecture of the wide context (WC) block. It was also first theoretically conceived by Mobeen Ur Rehman et al (2020) to combat various limitations in the U-Net architecture. Although similar to the RES block, the WC block has its own advantages. But before discussing that let us look at how the WC block is implemented. The WC block is placed right in the middle of the U-Net architecture and forms the block that separates the contracting and expanding pathways. The WC (Wide Context) in the architecture receives input from the first stage of the network and has two parallel connections, each with two convolution layers. In the first connection, the convolution layers use a combination of N x 1 and 1 x N filter sizes. In the second connection, the filter sizes are reversed, starting with a 1 x N filter and followed by an N x 1 filter.

The combination of filter sizes in both connections creates a robust feature set that contributes significantly to the performance. The outputs from both connections are then summed up and treated as the output of WC. The extracted contextual information from the WC, similar to RES, is essential in sub-classifying different sub-classes of cancer. Additionally, WC performs feature aggregation at the transition level, leading to a better reconstruction of segmented regions. However, the WC does not address information degradation or make the architecture scale invariant, unlike RES. As a result, RES outperforms WC significantly, as observed from the dice score, indicating that the disadvantage of WC affects its performance. [7]

3.3 RU-NET

As mentioned previously, after considering the superiority of RES over WC, we made the decision to utilize RES. Consequently, we proceeded to design a modified architecture by integrating RES blocks into the conventional U-Net framework. The implementation of this architecture was carried out in real time, employing a Google Colab notebook.

Google Colab serves as an online Jupyter notebook environment, offering the advantage of accessibility from anywhere without the need for installation. It provides a convenient platform where multiple team members can simultaneously edit the notebooks, similar to collaborative editing on a Google Doc. The versatility of Colab extends to its support for various popular machine learning libraries, which can be readily imported into the notebook for seamless integration. This feature allows for swift utilization of established machine learning tools within the Colab environment.

Our model training was accelerated with the use of a GPU hardware accelerator. Leveraging the free access to the "NVIDIA Tesla K80" GPU provided by the Google Colab environment, we harnessed the immense parallel computing power of the GPU to significantly improve the performance of our complex neural network-based model while reducing the training time. The GPU is equipped with numerous smaller cores, enabling massive parallel processing, and a high memory bandwidth, which was particularly advantageous for our task. In our case, using the CPU hardware accelerator tookus about two entire hours to be able to train even the basic U-Net for a single epoch during our preliminary experiments. But using the GPU has enabled us to train our hybrid model for an epoch under just eight minutes. Training for 30 epochs takes about 4 hours and doing the same for 50 epochs takes under 7 hours. Training time was one of our biggest impediments during the preliminary experimentation stage but this was overcome by using GPU. Furthermore, as our problem requires us to develop a model that performs best within a specified minimum number of epochs, we set up a constraint of allowing only 30 training epochs to bring out satisfactory results. We settled with the 30 epochs training period as the stricter the constraints the better the model's performance is highlighted.

For the implementation of our model, we chose to use the Keras library due to its ease of use, versatility, and consistency. Keras is an open-source Python library that provides a user-friendly interface for designing and training artificial neural networks. It functions as a front-end for the TensorFlow library, another open-source software library used for machine learning and artificial intelligence tasks, with a focus on deep neural network training and inference.

Keras includes a wide range of pre-built neural network components such as layers, objectives, activation functions, optimizers, and tools for handling image and text data, making it simpler to write code for deep neural networks. Furthermore, Keras supports popular neural network architectures like convolutional and recurrent neural networks, as well as frequently used utility layers like batch normalization, dropout, and pooling. Below is a diagram of the RU-Net model that we implemented using Keras.

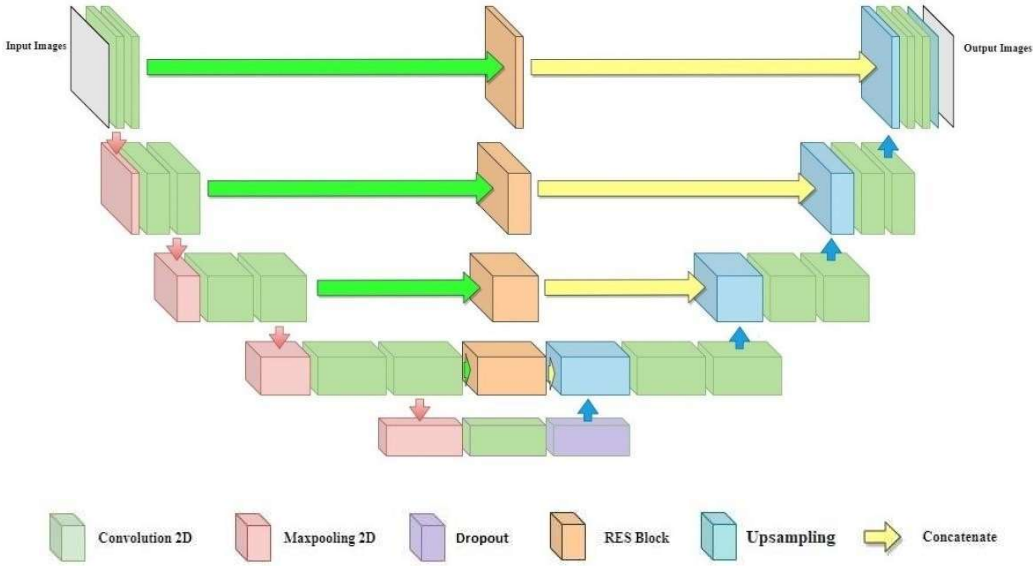

Fig. 3.3. Schematic visualization of the modified RU-Net architecture we have developed.

The Fig. 3.3. shows the resulting architecture that incorporates the RES blocks into the U-Net creating our modified architecture is the RU-Net. The various block of 2D convolutional layers that forms the basis of this architecture is shown in green colour. These were implemented using the Conv2D function. The Keras Conv2D is a type of layer used for 2-dimensional convolution, which involves applying a filter to the input data to generate a new tensor of outputs.

This filter is typically a small matrix that is slid over the input data to perform a mathematical operation at each position. In the implemented RU-Net, the pink blocks represent max pooling, which is a type of pooling operation that selects the maximum value from a specific area of a feature map, and creates a downsampled (or pooled) feature map. Max pooling is commonly used to reduce the spatial dimensionality of the output tensor while retaining the most relevant information.

It is implemented using the MaxPooling2D function which provides max pooling for 2D spatial data. The upsampling blocks are visualized in blue. The UpSampling2D function is used to implement this and it repeats the rows and columns of the data to expand it. Finally, the RES blocks are visualized using the orange blocks in the schematic. The entirety of the RES block as shown in Fig. 3.1. was implemented using Conv2D and concatenation functions in the code. The resulting RU-Net architecture gives the maximum possible dice coefficient values for all the tumor clusters we do segmentation for within the problem constraints. We have finalized this architecture, experimented with it and tabulated the results in the next chapter.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter we will compile the results of all our experiments and compare the performance of the standard U-Net and WC infused U-Net against our RU-Net architecture.

4.1 PRELIMINARY DATASET VISUALIZATION

In this section we will take a look into the dataset that we have to work with, namely the BraTS dataset, and the data that available in it.

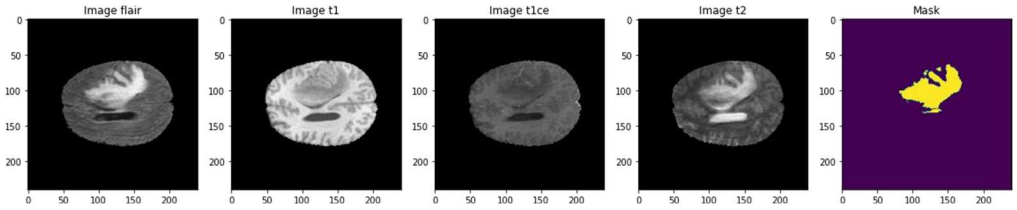

Fig. 4.1. Tumor image from BraTS dataset.

The Fig. 4.1 provides us a look at the tumor, from a BraTS dataset MRI scan, as shown by the different MRI modalities. From left to right: FLAIR (Fluid-Attenuated Inversion Recovery), T1, T1 contrast-enhanced, T2 and a masked image highlighting the whole tumor region are visualized here.

The next visualization Fig. 4.2. displays a layered view of the 3D MRI scan. It shows all the different slices which composes a scan from the BraTS dataset, thus making it a 3D MRI. This allows segmentation algorithms to completely map out and learn the tumor regions thus improving accuracy.

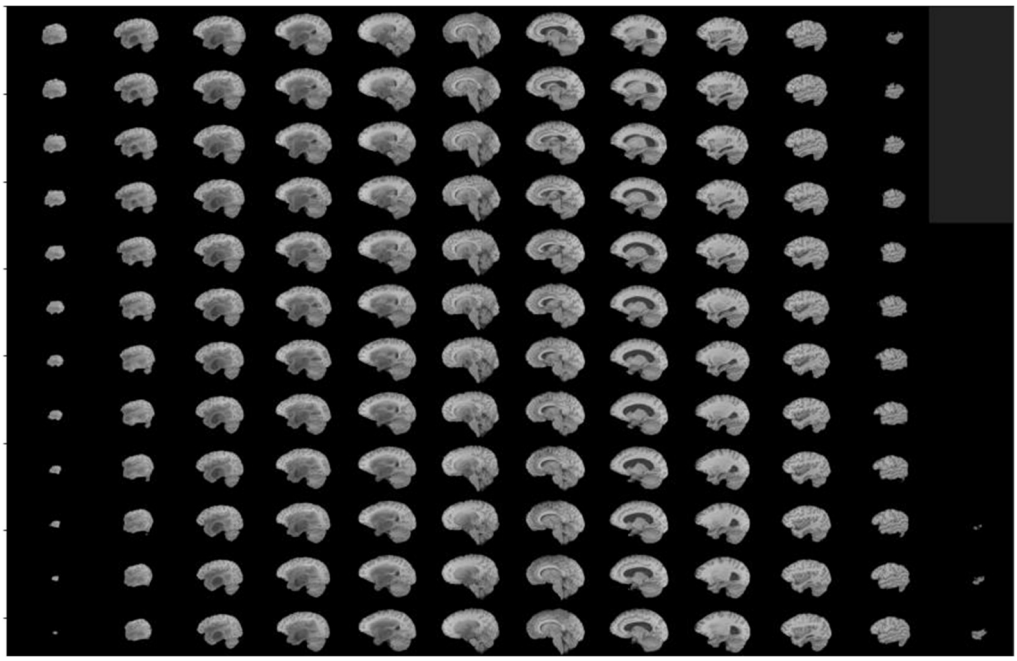

Fig. 4.2. A layered view of an MRI scan.

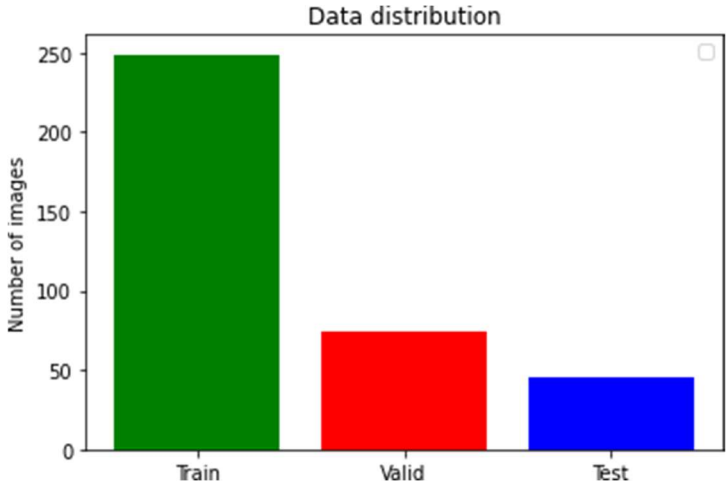

Fig. 4.3. Train-validation-test split of the MRIs from the BraTS 2020 dataset.

The above figure, Fig. 4.3., shows the data from the BraTS 2020 dataset split up for training, validation and testing. split of the MRIs. These

values of train-validation-test split were finalized after experimentation and via suggestions from the literature. [4][9]

4.2 PERFORMANCE METRICS

The metrics that we use to analyze the performance of our model are listed below:

1) Accuracy: Accuracy is a metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{4.1}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.2}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

However, accuracy alone doesn't tell the full story when you're working with a class-imbalanced data set, like the BraTS dataset we're dealing with now, where there is a significant disparity between the number of positive and negative labels. We'll now look at two better metrics for evaluating class-imbalanced problems: precision and sensitivity.

2) Precision: Precision tells us what proportion of positive identifications was actually correct. Precision is the ratio of correctly predicted positive

observations to the total predicted positive observations.

Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4.3}$$

where TP = True Positives, and FP = False Positives. High precision relates to the low false positive rate. Hence, precision helps us to visualize the reliability of the machine learning model to provide us correct positive classification.

3) Sensitivity: Sensitivity tells us what proportion of actual positives was identified correctly. Sensitivity is the ratio of correctly predicted positive observations to the all observations in actual class.

Mathematically, sensitivity is defined as,

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{4.4}$$

where TP = True Positives, and FN = False Negatives. Hence, sensitivity measures the model's ability to detect positive samples.

4) Specificity: Specificity is defined as the proportion of actual negatives, which got predicted as the negative.

It is mathematically defined as,

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{4.5}$$

where TN = True Negatives, and FP = False Positives. High specificity means that the model is correctly identifying most of the negative results, while a low specificity means that the model is mislabeling a lot of

negative results as positive. Ideally, the model would be expected to have a very high specificity or true negative rate as it is robust measure of the model's reliability towards in the field of risky decision making.

5) Dice Coefficient: Dice coefficient or dice score is the definitive metric for analyzing the performance of segmentation. It was used by the previous state-of-the-art techniques, so it will allow us to have a better quantitative comparative analysis between existing state-of-the-art and proposed RU-Net architectures.

The Dice similarity coefficient, also known as the Sørensen–Dice index or simply Dice coefficient, is a statistical tool which measures the similarity between two sets of data. This index has become arguably the most broadly used tool in the validation of image segmentation algorithms. The Dice score gives similarity between sets P & Q which can be mathematically expressed as,

$$\text{Dice} = \frac{2\times|P\cap Q|}{|P|+|Q|} \tag{4.6}$$

where |P| and |Q| represents the cardinalities of sets P & Q respectively.
It is also expressed as,

$$DSC = \frac{2TP}{2TP+FP+FN} \tag{4.7}$$

where TP = True Positive, FP = False Positives, and FN = False Negatives.

Now that we have a good understanding of all the metrics, we will start discussing the experiments we conducted to arrive at the optimal model.

4.3 OPTIMAL U-NET EXPERIMENT

Since the basis of our architecture is the U-Net we begin by experimenting with it. We create different simple U-Net architectures of varying depths namely, 18, 23 and 28 layers. The models were later tested and then the performance parameters are tabulated.

Table. 4.1. Shown here is the mean performance metrics of simple U-Net models of varying depths on the BraTS 2020 dataset.

Parameters	18 layer U-Net	23 layer U-Net	28 layer U-Net
Accuracy	0.9831	0.9837	0.9836
Precision	0.9847	0.9838	0.9834
Sensitivity	0.9818	0.9836	0.9832
Specificity	0.9949	0.9946	0.9945
Dice Coefficient	0.2862	0.2986	0.2853

No. of epochs: 2

The above table, Table. 4.1., is the tabulation of the mean performace parameters of simple U-Net models of varying depths on the BraTS 2020 dataset. Clearly the standard 23-layer U-Net is superior. Though the whole tumor regions were easily predicted leading to high accuracy values, due to the number of training epochs being just two, the hardware constraints, namely using a CPU instead of a GPU for training, and due to the model's architecture being simplistic, the segmentation task

of identifying and demarcating tumor subregions was poorly done leading to low dice coefficient values. Therefore, since every other parameter has already attained very good values, the focus from now on will be solely increasing the dice values as they're the most important metric that gives us information about the segmentation performance. Hence, we shall from now on record only the accuracy, precision and dice scores.

4.4 ORIGINAL VS PREDICTED TUMOR REGION VISUALIZATION

1. The following are the outputs obtained for the simple U-net architecture trained for 2 epochs:

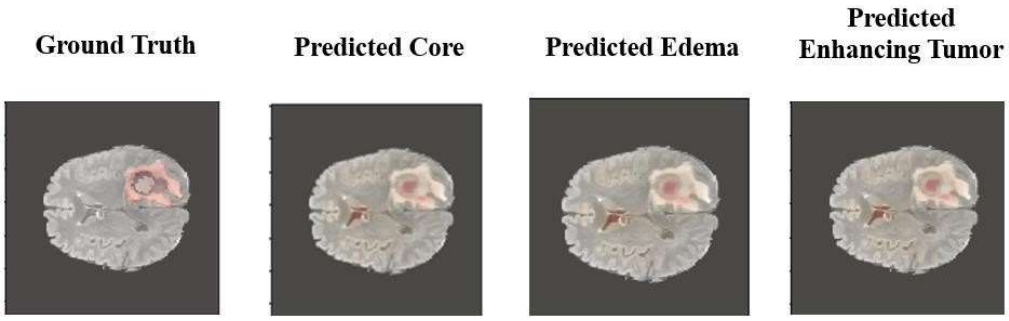

Fig. 4.4. Sample outputs of the simple U-Net trained for 2 epochs.

As demonstrated by these sample outputs, though the tumor seems to be detected with high accuracy the subregions are grossly misidentified as shown by the discrepancies between shading in the actual and predicted output images. This is due to the very low of training epochs. Thus, we increase the number of epochs to 30 to increase the segmentation accuracy.

2. The following are the outputs obtained for the simple U-net architecture trained for 30 epochs:

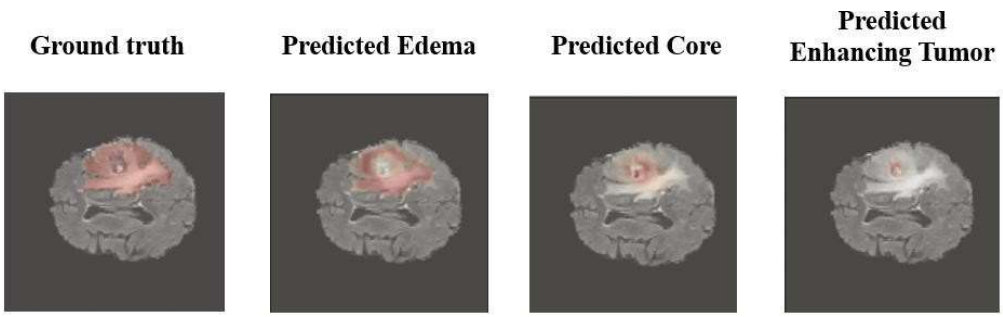

Fig. 4.5. Sample outputs of the simple U-Net trained for 30 epochs.

As seen in the above samples, with an increase in number of epochs the accuracy in predicting subregions has remarkably increased. However, on close inspection of the images we see that the model is struggling to differentiate between the core and the enhancing parts of the tumor. This is due to the informational loss in the U-Net pathways and the imbalance in the dataset due to the nature of brain tumor MRIs. [5] [7] Thus, we now implement the RU-Net to nullify these problems.

3. The following are the outputs obtained for the modified U-net architecture with a WC block trained for 30 epochs:

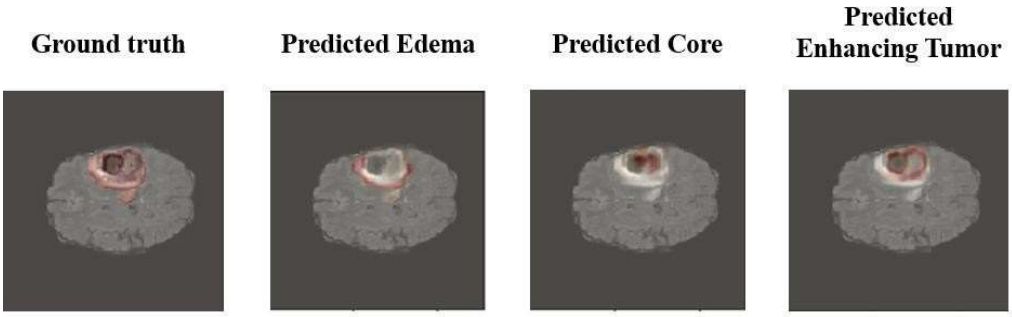

Fig. 4.6. Sample outputs of the modified U-Net with WC block trained for 30 epochs.

There is some improvement in the output when the WC block is implemented but the problem of differentiating between the core and

enhancing tumor still does persists as WC doesn't address the informational loss and dataset imbalance.

4. The following are the outputs obtained for the RU-Net (modified U-net architecture with RES blocks) trained for 30 epochs:

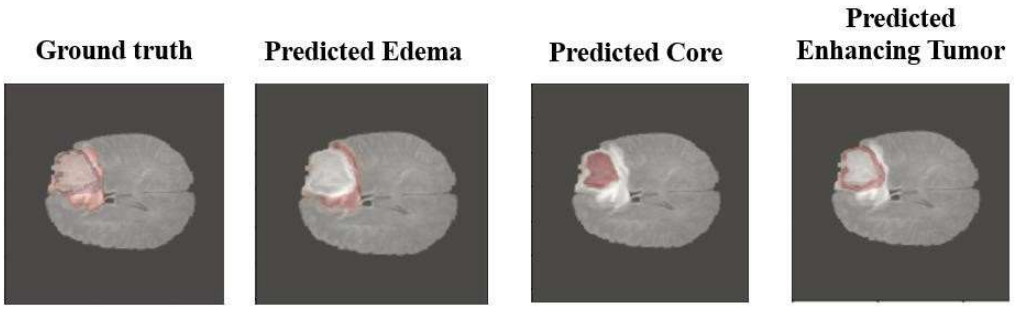

Fig. 4.7. Sample outputs of the RU-Net trained for 30 epochs.

As seen in the above samples with added RES blocks to modify the architecture the accuracy in predicting subregions has increased to a very high degree. Unlike before the demarcation between the enhancing tumor region and the core of the tumor is now very clear and is fairly devoid of errors. Thus, we obtain the best segmentations by utilizing RU-Nets.

Now that we have seen the visualizations of the segmentation outputs let us move to inspecting the performance parameters of all these different architectures.

4.5 TEST RESULTS FOR PERFORMANCE METRICS

1. Simple U-Net architecture for 2 epochs:

Table. 4.2. The dice coefficient and other performance metrics of the simple U-net model on BraTS 2020 dataset for two epochs training.

Parameters	Values
Accuracy	0.9837
Precision	0.9838
Dice Coefficient	**0.29**

No. of epochs: 2

As seen here the dice score of the simple U-Net trained for just two epochs is very low. This situation is perfectly illustrated by Fig. 4.4 wherein there was practically no segmentation of the tumor regions. Now, to increase the dice score the number of epochs is increased from 2 to 30, which is our epoch limitation constraint as discussed previously. The resulting changes in metrics are shown below.

2. Simple U-Net architecture for 30 epochs:

Table. 4.3. The dice coefficient and other performance metrics of the simple U-Net on BraTS 2020 dataset for 30 epochs training.

Parameters	**Values**
Accuracy	0.9924
Precision	0.9935
Dice Coefficient	**0.6248**

No. of epochs: 30

As expected, the increase in the number of training epochs has vastly increased the dice score, meaning the segmentation capability of the model has been greatly increased. Now, to increase the dice score the WC and the RES architectures are both implemented and their respective dice scores are compared to see which one is superior.

3. U-Net architecture modified with WC block:

Table. 4.4. The dice coefficient and other performance metrics of the U- Net modified with WC block on BraTS 2020 dataset for 30 epochs training.

Parameters	**Values**
Accuracy	0.9929
Precision	0.994
Dice Coefficient	**0.6462**

No. of epochs: 30

We see here that the modified U-Net with the WC block doesn't give a good performance boost as far as the dice coefficient is concerned. This is due to the fact that the biggest antagonists to the dice score are the dataset imbalance and the informational loss that occurs in the U-Net. To handle both of these issues, the RU-Net utilizes the RES blocks.

4. The optimized RU-Net architecture:

Table. 4.5. The dice coefficient and other performance metrics of the optimized RU-Net on BraTS 2020 dataset for 30 epochs training.

Parameters	**Values**
Accuracy	0.9934
Precision	0.9941
Dice Coefficient	**0.72** (Range: 0.6879 – 0.7493)

No. of epochs: 30

The optimized modified U-Net with RES blocks, namely the RU-Net, gives the best dice coefficient values on both the training and test dataset and outperforms all the other architectures in segmenting brain tumors.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In conclusion, the study on Brain Tumor Segmentation utilized an optimized RES block infused RU-Net architecture which performed exceptionally well compared to other real-time large-scale architectures when deployed to segment brain tumors. The RU-Net outperformed all other architectures when trained for just 30 epochs due to its ability to prevent information degradation and make the architecture scale invariant. The RU-Net architecture obtained dice scores of 0.7234, 0.7172, 0.7863, and 0.8557 for the whole tumor, necrotic, enhancing, and edema subregions, respectively. The developed model can assist neurologists, oncologists, and surgeons in tumor diagnosis, prognosis, and surgery by accurately identifying and highlighting various tumor subregions.

To improve the RU-Net architecture, it is recommended to scale up the hardware resources, increase the number of epochs, and adjust other parameters.

Additionally, creating an ensemble using the developed modified model is an interesting direction for future studies. By using the same principle, an ensemble can be created that performs segmentation more accurately and swiftly. In addition, we can use pretrained weights for an autoencoder in the future to optimize performance. By using a pre-trained autoencoder, we can initialize our deep learning model with weights that have already learned useful features from the dataset, thereby improving the efficiency of training and potentially improving the accuracy of the model.Overall, the RU-Net architecture has great potential in the future of brain tumor segmentation and can greatly benefit the medical field.

REFERENCES

[1] Bakas S., Akbari H., Sotiras A., Bilello M., Rozycki M., Kirby J.S., et al. (2017), "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", Nature Scientific Data.

[2] Bakas S., Reyes M., Jakab A., Bauer S., Rempfler M., Crimi A., et al. (2018), "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge".

[3] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M. and Larochelle, H., (2017). Brain tumor segmentation with deep neural networks. Medical image analysis, 35, pp.18-31.

[4] Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D. and Glocker, B., (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Medical image analysis, 36, pp.61-78.

[5] Menze B. H., Jakab A., Bauer S., Kalpathy-Cramer J., Farahani K., Kirby J., et al. (2015), "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)", IEEE Transactions on Medical Imaging 34(10).

[6] Myronenko, A. (2018), September. 3D MRI brain tumor segmentation using autoencoder regularization. In International MICCAI Brainlesion Workshop (pp. 311-320). Springer, Cham.

[7] Rehman MU, Cho S, Kim JH, Chong KT., (2020), BU-Net: Brain Tumor Segmentation Using Modified U-Net Architecture. Electronics; 9(12):2203.

[8] Ronneberger, O., Fischer, P. and Brox, T., (2015), October. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241).

[9] Wang, F., Jiang, R., Zheng, L., Meng, C. and Biswal, B., (2019), October. 3d u-net based brain tumor segmentation and survival days prediction. In International MICCAI Brainlesion Workshop (pp. 131-141).