3D Brain Image Segmentation Using 3D Tiled Convolution Neural Networks

by

Md Mahibul Haque

20101503

Jobeda Khanam Ria

20101217

Fahad Al Mannan

20101155

Sadman Majumder

20101224

Md Reaz Uddin

20101228

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2023

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md Mahibul Haque
20101503

Jobeda Khanam Ria
20101217

Fahad Al Mannan
20101155

Sadman Majumder
20101224

Md. Reaz Uddin
20101228

Approval

The thesis/project titled "3D Brain Image Segmentation Using 3D Tiled Convolution Neural Networks" submitted by

1. Md Mahibul Haque (20101503)
2. Jobeda Khanam Ria (20101217)
3. Fahad Al Mannan (20101155)
4. Sadman Majumder (20101224)
5. Md Reaz Uddin (20101228)

of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September, 2023.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Ashraful Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

We hereby declare that this thesis is our own work and the discoveries are from our research. The sources have been appropriately acknowledged which we used in our study. Therefore, we are confirming that this thesis has not been submitted published, or presented in any other educational institution for receiving any degree.

Abstract

Gliomas are the primary brain tumors that are most commonly observed in adult patients and exhibit varying degrees of aggressiveness and prognosis. The accurate identification and diagnosis of Gliomas in surgical procedures heavily rely on the acquisition of precise segmentation results, which involve delineating the tumor region from magnetic resonance imaging (MRI) scans of the brain. The segmentation process in conventional 3D CNN methods is often reliant on patch processing as a result of the limitations in GPU memory. This paper presents an approach for segmenting brain tumors into distinct subregions, namely the WT, TC, and ET, utilizing a 3D tiled convolution-based segmentation method. The utilization of the 3DTC method enables the inclusion of larger patch sizes without requiring hardware with high GPU memory. This study presents three significant modifications to the standard 3D U-Net. Firstly, we incorporate 3D tiled convolution as the initial layer in our proposed models. Secondly, we substitute the trilinear upsampling layer with a dense upsampling convolution layer. Lastly, we replace the standard convolution block with recurrent residual blocks in the proposed R2AU-Net. The best framework was utilized to apply an average ensembling technique, aiming to achieve accurate results on the validation set of the BraTS 2020 dataset. The network proposed in this study was utilized for the analysis of the BraTS 2020 dataset. The evaluation of our method on the validation dataset yielded Dice scores of 90.76%, 83.39%, and 74.77% for the WT, TC, and ET regions, respectively.

Keywords: Deep learning; 3D tiled convolution; MRI; Segmentation

Acknowledgement

We are heartily grateful to Almighty Allah, who made it possible to finish our Thesis for us on time without any major obstacles, all praise for our almighty creator. After that, we would like to confess our deepest appreciation to Dr. Md. Ashraful Alam, our supervisor, whose unconditional encouragement made it possible for us to finish the project. Sir helped us in every possible way we needed for the research; such as Sir allowed us to use the CVIS Lab, which was an invaluable resource throughout the testing and documentation phases of our project. We are thankful to our honorable assistant professor. We are also thankful to Dr. Golam Rabiul Sir, for giving us the chance of using a high configuration PC from the Research lab.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The use of convolutional neural networks (CNNs) to problems in computer vision and medical image processing has increased recently [1]. In recent years, CNN has found a huge usage in medical image processing methods. Despite this extensive use, most methods only support processing 2D images, even though 3D images make up the overwhelming majority of recorded medical imaging data. There are many different kinds of 3D imaging modalities, such as MRI, CT, and micro CT scanners, and 3D image segmentation is used to find and extract relevant regions from the generated datasets. With the development of 3D CNN technology, they will have access to greater geographical context [1], [2]. Because of its complex structure—often including many nonlinear processes and many trainable parameters—To extract meaningful visual representations from pictures, Convolutional Neural Networks (CNNs) may be very helpful [3]. Most 3D image analysis relies on segmented images, while real-time measurement and analysis of 3D images is possible in certain cases. Scanned items may be measured and seen with the use of 3D models, which can be transformed utilizing 3D image segmentations to extract the geometry of an area of interest. By completing the segmentation of the 3D pictures, virtual inspections of these models may be performed by means such as computer simulations or the building of a physically represented problem through 3D printing. In [4], we see an examination of the ways in which medical images may be processed using graphics processing units (GPUs), including procedures like segmentation, visualization, and registration. Many computationally intensive applications may now be greatly accelerated because to the GPU's increased processing power, which was highlighted in contrast to older GPU-based computing frameworks. Recently, GPU has become a competitive platform due to its massive processing capability, which has allowed for improved computing performance [4]. Because of the GPU memory constraints introduced by a full conversion to 3D, modern techniques depend on patch-processing and sub-volumes. Even with a powerful graphics processing unit (GPU) and no specialized hardware, the size of the patch to be entered is often rather tiny, making it difficult to synthesize large amounts of contextual information at a reasonable pace [5]. For our 3D Semantic Segmentation problem, we propose using 3D Tiled Convolution to address the loss of contextual information and reduce the computational cost.

1.2 Motivation

With the advancement of AI, more people hope to use AI approaches for the automated segmentation of tumors in MRI datasets. CNNs are a specific kind of ANN utilized in deep learning for the purpose of interpreting pictures. Some examples of these applications are IP [6], classification or segmentation, BCI, financial series data, and picture and video recognition. Convolutional Neural Networks (CNNs) may be regarded as a notable instance of a distinct kind of deep learning model, as highlighted in [7]. According to [8], there are segmentation techniques that may directly manipulate the original dataset [9]. CNN has been used before to facilitate 2D computer vision tasks and to do image processing for medical applications. CNN is used before for the purpose of driving 2D CV and performing tasks associated with the processing of medical images [8]. Given that a significant proportion of clinical pictures now accessible are obtained in three-dimensional (3D) format, there exists a compelling impetus to continue advancing the development of three-dimensional convolutional neural networks (3D CNNs). This is primarily driven by the need to harness the supplementary spatial context offered by such networks. Two-dimensional convolutional neural networks (CNNs) are used to predict segmentation maps for single-plane anatomical MRI scans. While 3D-CNNs solve this problem by using 3D convolutional kernels to predict segmentation for volumetric areas of a scan, this kind of network is still in its infancy. It is generally agreed that fully convolutional networks represent a superior class of models capable of performing a wide range of pixel-level tasks. FCNs mimic the behavior of fully linked layers as CNNs do but lack depth layers. As an alternative, they use 1 x 1 convolutions [10]. Recently, FCNs for semantic segmentation have seen significant accuracy improvements thanks to the transfer of pre-trained classifier weights, layer representations, a fusion of several models, and end-to-end learning on whole image volumes data sets. To predict semantic segmentation without spatial data disagreement and to identify illnesses circling around the area, we may use 3D-TC in conjunction with efficient 3D FCNs in volumetric MRI images. We provide a method for 3D segmentation that makes use of 3D-TC on volumetric pictures to get optimal results. Our aim is to deliver a significant decrease in the computation cost associated with 3d Medical Image Processing and improve overall segmentation performance with the use of 3D Tiled Convolution & 3D Fully Convolutional Networks.

1.3 Problem Statement

Obtaining accurate volumetric images of a patient's internal organs is essential for medical professionals when establishing a diagnosis or identifying an abnormality. Diagnostic and therapeutic success rely heavily on the accurate segmentation of volumetric images such as MR (Magnetic Resonance) scans [11].Analysis of medical images using digital image analysis and other imaging modalities is a growing field of study. However, a quick and effective surgical treatment is ensured by very accurate segmentation, which aids in diagnostic and surgical planning. [5]. Simply said, it ensures higher quality medical attention. Inter-patient anatomical heterogeneity makes it difficult to automate volumetric medical visual segmentation, since different organs have different sizes, forms, and architectures. We had to find a way to fix the evident accuracy issue without causing any patients' therapy to be stalled out

for too long. This necessitated a very efficient segmentation technique, which could be achieved by the deployment of a powerful 3D U-net model. Segmentation of 2D medical images is well established by CNN. [11]. However, because to the additional complexity of 3D volumes and the variety in organ size and shape, segmenting 3D image volumes remains a challenging problem.
As a result, 3D Fully Convolutional Networks may be used to successfully segment the volumetric images and address these concerns. However, the high computational cost of implementing a 3D FCN is a serious drawback. Unless specialized hardware with a lot of GPU RAM is used, the input patch's size maybe limiting in 3D FCNs, making it difficult to provide more context information for improved performance. The computational load may be lightened while the spatial context is preserved using 3D Tiled Convolution (3D-TC). Many different kernels are learnt in the same layer using 3D Tiled Convolution (3D-TC). The processing time for 3D medical image volumes is not significantly impacted by the reduced amount of GPU RAM required by 3D-TC. In this research, we propose an ensembled 3DTC segmentation architecture for performing efficient brain tumor segmentation.

1.4 Research Objectives

The main objective of our research is to learn about the 3D Tiled Convolution on segmenting volumetric Brain images. More spatial context is often included when using 3D CNNs for volumetric picture segmentation as opposed to the more common 2D CNNs method. 3D Convolutional Neural Networks require substantial GPU memory which becomes huge restraint. This study proposes 3D-TC that greatly lowers the GPU memory consumption involving 3D medical image processing. The objectives of this study are:

1. To deeply evaluate 3D-TC, how it impacts the performance in semantic segmentation .

2. To evaluate popular 3D semantic segmentation architecture.

3. To develop a model that incorporates the 3D-TC approach along with the 3D segmentation architectures.

4. To provide suggestions about how the model can be made more accurate and how 3D tiled convolution may be used to better segment 3D data in the future.

Chapter 2

Related Work

Each year, there are about 60 million MRIs conducted globally among which about 12 million MRIs are conducted on the brain to diagnose any anomaly. One of the main purposes of the MRI is to detect regions of the tumor and categorize the region with its distinctive importance. To analyze brain MRI, image segmentation is commonly utilized for surgical planning, image-guided procedures, assessing brain changes, measuring and visualizing the anatomical features of the brain, and defining diseased areas. When it comes to image segmentation and other forms of visual identification, deep convolutional networks have made great strides in recent years. Although CNNs have been available for some time, their usefulness has been limited by the limitations of both the training sets and the evaluation networks. The suggested network in [3] evaluated the issue and found a solution: it has two paths. Where shortening strategy used standard layout in two 3x3 convolutions. The researchers used an overlap-tile method to successfully split the massive picture. Several biological segmentation tasks have been completed using the 2D U-net-based network.

Exploring the well-established 2D U-net published in [3], a 3D U-net design was introduced on [12] in 2016. In this design, all 2D tasks are replaced by their corresponding 3D equivalents. The proposed network was developed specifically for volumetric image segmentation, and it is meant to be trained with just a few labels applied to the training images. Both a semi-automated configuration in which certain slices of the volume to be segmented have been annotated and a fully-automated design in which the network is expected to learn from existing sparsely-annotated volumes are covered in this study. The suggested networks were built with volumetric image segmentation in mind, and it was designed to be able to learn from minimally labeled volumetric imagery during training. Some slices of the volume to be segmented have been annotated, as in the semi-automated configuration addressed in this study; in the fully-automated setup, the network is intended to learn from existing sparsely annotated volumes included in the training dataset. This building benefited greatly from the addition of volumetric photos since they provided crucial contextual information about the surrounding environment that had been lacking in previous, 2D methods. Experiments utilizing the network suggested that a completely automated setup performed better when presented with a bigger sample size. In [1], a volumetric full CNN is presented as a means of 3D image segmentation. As part of the training process, researchers established a unique goal

function based on the DSC and then optimized it to maximize training performance. This allowed for the handling of cases when the ratio of foreground to background voxels was very high or low. It was broken up in many phases, could run at a higher or lower resolution. Volumetric kernels of size $5 \times 5 \times 5$ voxels were employed in all convolutions throughout all stages. By distributing the network workload over numerous GPUs, we can build the necessary scale to divide volumes into various regions.

A 3D densely convolutional network was designed to maintain multi-scale contextual information [13]. The suggested network design was built on the existing DenseNet framework, which allowed for a seamless transition across network levels. Through the combination of fine and coarse dense blocks in feature maps, the suggested network design enhanced the flow of information. The suggested architecture combines fine and coarse dense blocks of feature maps to gather multi-scale contextual information for integrating local forecasts with global projections. After comparing with the 3D U-net design having 18 layers and 19 million learned parameters, this network architecture achieved 92.50% with a smaller amount of parameters while still performing adequately. During testing, the 3D U-net architecture performed at 91.58% accuracy.

Segmentation designs have traditionally included a DS route, coarse semantic, an US path trained restoring input picture resolution at the model's output, and a post-processing module optimize model predictions. Densely Connected Convolutional Networks (DenseNets) are a revolutionary CNN design that has recently shown outstanding results in imagery classification. DenseNets are founded on the premise that a network will be more accurate and simpler to train if every layer has a direct feed-forward connection to every other layer. The network presented in [14] makes use of an upsampling route to restore the entire input resolution in the output, allowing it to function as an FCN. According to the proposed network architecture, the upsampled dense block integrates data from many dense blocks between the two paths, and a regular skip connection carries the higher-resolution data. Since a fully convolutional DenseNet has fewer parameters than most other segmentation topologies, it naturally benefits from (1) parameter efficiency. As opposed to a simple extension of DenseNets, where feature maps grow linearly, the upsampling method reduces size which showed a very complex network with great depth, with 56 to 103 layers but few distinguishing features. DenseNet, the recommended network, includes drawbacks such as data redundancy and layer-mixing feature maps. The exponential development in compute and memory cost during training occurs because the number of model parameters increases exponentially with the network layers. Therefore, if there are too many layers inside the network, the entire model's performance suffers.

A different dual-pathway-based network which was 11 layers deep, incorporating a 3D convolutional neural network was proposed in [15]. Since class imbalance is a problem common to biomedical datasets, the suggested network was able to automatically adjust to the data. Soft segmentation 3D fully linked CRF was utilized which helped in segmentation. Pathways provided local features at high resolution

and the second pathway provided global features at low resolution. After several iterations of convolution, the network's residual connections and global and local paths were combined. Another difficulties from 3D CNNs was alleviated by this network, which demonstrated the analysis of massive visual context employing in multi-scale processes.

The authors of [16] presented an architecture called 3D Dense U-Net for fully automated high-resolution 3D volumetric segmentation of medical imaging data using supervised deep learning. The proposed network's implementation was modified to optimize for GPU-accelerated high-resolution image processing, enabling direct processing of three-dimensional images. The 3D Dense-U-net model demonstrated superior performance on the brain MRI dataset, achieving an accuracy of 99.72%. This result surpasses the performance of the previously disclosed 3D U-net architecture. The use of a 3D U-net extension has the potential to significantly enhance the performance of semantic segmentation tasks. The suggested separable 3D U-Net architecture in [17] utilizes individual 3D convolutions for separation purposes. The "S3D-UNet" network used for brain segmentation effectively utilizes the 3D volumes by using a unique separable 3D convolution approach. This approach involves dividing each 3D convolution into three parallel branches. The proposed network utilizes a convergent pathway and a divergent trajectory, similar to that of an auto-encoder.

In [18], a novel unsupervised segmentation algorithm was proposed for 3D medical pictures. Since the introduction of convolutional neural networks (CNNs), image segmentation has made tremendous strides. However, most existing methods focus on supervised learning, which calls for a massive quantity of labeled data. As a consequence, the ever-increasing volume of medical images overwhelms conventional approaches. Together, clustering and unsupervised deep representation learning are offered here as a unified approach to segmentation. JULE's ability to analyze 3D medical images is thanks to the incorporation of 3D convolutions into the architecture of CNN. Here, we work on biomedical picture segmentation using JULE and k-means. Since there is room for error in MRI, it is helpful that JULE can learn characteristics that distinguish between regions of high and low intensity.

An additional noteworthy 3D AlexNet Network-based automatic segmentation setup was suggested in [19]. When compared to ResNet50 and Inception-V4, the proposed 3D AlexNet network performed very well in terms of the time and number of parameters required for training. The 3D AlexNet network was presented as an extension of AlexNet, which was first proposed by Krizhevsky et al. [20]. The input format for this model was $227 \times 227 \times 3$, and it had 8 layers. Convolutional neural network AlexNet was proposed, and it employed a residual connection with pReLu activation. The 3D AlexNet approach utilized for the automated classification of prostate cancer MRIs minimized the number of network layers and channels, which in turn minimized the number of model parameters and shortened the training time. Prostate cancer may be effectively diagnosed using the proposed 3D AlexNet network, which achieved a 96.40% accuracy rate when classifying prostate tumors as benign or malignant using MRI scans. At this point, we may conclude that AlexNet can serve as a foundation for training on more extensive data sets or across many modalities.

The framework known as "No-new-Net," or nnU-Net, is a resilient and adaptable system built around the fundamental principles of vanilla U-Nets in both two and three dimensions [21]. The approach relies on a set of three U-Net models, which exhibit a very simple structure and make minor adjustments to the original U-Net architecture proposed by Ronneberger et al. (2015). The neural network designs of nnU-architectures Net are dynamically adapted to match the specific geometry of the input picture. Commencing with the network configurations that were previously determined to be compatible with our technology, our objective was to construct a network from its foundational elements. The input patches have dimensions of $256x256$. There are a total of 42 batches. In the 2D U-Net architecture, the deepest layers consist of 30 feature maps. It is worth noting that the number of feature maps increases twofold with each downsampling step. Similar to its 2D counterpart, the 3D U-Net used 30 feature maps in the layers with the highest resolution. In this particular instance, the original setup included a batch size of 2, with the input patch size being 128x128x128. The authors proposed the use of the MSS U-Net technique, as documented in [22], to segment kidneys and renal tumors via the analysis of CT images.In order to improve the effectiveness of the 3D U-training Net, the suggested framework integrates both deep supervision and exponential logarithmic loss. The proposed network is derived from the well-recognized nnU-net design for neural networks. The network architecture that was put forth had six discrete layers. The depth of feature maps does not exceed an 8x8x8 grid. Concurrently, the volume of the model was decreased by setting the fundamental kernel number to 30. In comparison to more complicated structures, which may have inflated models and poor reproducibility, the advantages of this streamlined design include greater repeatability and broader applicability of findings.

To efficiently execute both learning and inference from beginning to finish, the proposed 3D DSN makes use of a fully convolutional architecture [11]. To overcome possible optimization challenges during training, a deep supervision mechanism was included; as a result, the suggested model was able to achieve a much quicker convergence rate and stronger discriminating capabilities. The proposed DSN was built considering complete three-dimensionality for optimal encapsulation of spatial information in the volumetric data. While the 3D DSN may provide high-quality probability maps, simply thresholding the probabilities is not necessarily a reliable technique to identify the contour of unclear areas. By using a contour refinement CRF model, the segmentation results were made even more accurate.

In [23], it was proposed to automatically search for a 3D segmentation network utilizing C2FNAS. The search was carried out in two phases: the first, a coarse phase, concentrated on the topology of the network as a whole, or how each convolution module is connected to other modules; the second, a fine phase, concentrated on the micro-level and sought out operations in each cell by using the network's topology as a starting point. Due to the memory-intensive nature of 3D segmentation, traditional NAS techniques often fall short when tasked with segmenting 3D medical images; nevertheless, the proposed C2FNAS automatically generates a transferrable 3D segmentation network.

Tiled convolutional neural networks (CNNs) provide the computational flexibility

to learn different uniforms while also recognizing the value of basically minimizing the number of absorbable boundaries. This study details an unaided pretraining approach, based on a tweaked version of Topographic ICA, for learning how to recognize and locate the various features of a geographical reference work (TICA). TICA may be utilized effectively to pre-train Tiled CNNs utilizing local symmetry, as demonstrated by [24] which showed how to group similar feature sets. The subsequent application of Tiled CNNs and their mention in the TICA study demonstrates that these networks are indeed well-equipped to learn uniform representations and feature pooling units that are resilient to scaling and rotation. Tiled CNNs were able to take seriously the newly distributed findings on the NORB and CIFAR-10 datasets when they found them, improving order execution in the process. Furthermore, the aforementioned studies demonstrated the efficacy of CNNs in performing several different types of acknowledgment tasks. Examples include mutual recognition of numbers from the MNIST dataset, recognition of objects from the NORB dataset, and the establishment of shared vocabularies. The outcomes proved the worth of reducing pressure throughout the order execution process. Pre-trained networks with 2.5 million unlabeled pictures from the small image dataset revealed a presentation increase from k = 1 to 3 and a plateau at k = 4. It was simple to prepare the Tiled CNN using any unlabeled data that was already available. Based on these results, it seems that mounting k = p is a viable option wherever there is sufficient data to prevent overfitting. In addition, Tiled CNNs were presented as a development of CNNs that support unsupervised pretraining and weight tiling.

The advancement of semantic segmentation in volumetric images was significantly enhanced by the introduction of a novel Holistic Decomposition Convolution method, as shown in the study conducted by the authors in [25]. This approach is effective and cost-efficient. The implementation of the suggested approach resulted in a substantial reduction in the amount of raw data necessary for further processing. The latest approaches using three-dimensional convolutional neural networks (3D CNNs) use a convolutional structure that involves downsampling and upsampling. The downsampling technique used in this method has a notable spatial resolution, and its primary objective is to gradually reduce the resolution of low-level features, hence facilitating the enhancement of feature abstraction. As stated in the aforementioned article, the diminishment of substantial contextual information arises due to the utilization of small input patch sizes in these procedures. To get volumetric dense prediction at the final output, the recommended approach utilizes HDC (Hierarchical Dilated Convolutions) in conjunction with sub-sequential CNNs (Convolutional Neural Networks). The proposed network then recovers its full resolution by using dense up-sampling convolution. The proposed approach shows enhanced efficacy by the integration of HDC and DUC with 3D U-net, HighRes3DNet, and 3D V-net. A high-dimensional convolution (HDC) was used as a cyclic down-shuffling operator followed by a low-resolution convolution, while a downsampling and upsampling convolution (DUC) was developed as the inverse of the HDC. The HDC algorithm generated a set of k feature maps. The performance of the 3D U-net was shown to be superior when using a larger patch size. However, due to limitations in GPU memory, we were constrained to a patch size of $200 \times 200 \times 40$. The dimensions of the patch for the 3D LP U-net were consistently set at 400x400x80. Various shuffling factors were tested, and the optimal results were obtained with a shuffling

factor of (4, 4, 2). The inclusion of a larger patch in the 3D LP-U-net model resulted in significant benefits, as seen by the considerable flattening of the learning curves. Despite having the fewest training parameters, the High-requirements ResNet exhibited the largest demand on GPU RAM to accommodate intermediate outcomes. The LP-U-net, V-net, U-net, and HighRes3DNet achieved average Dice Overlap Coefficients (DOC) of 96.76 ± 0.92%, 94.01 ± 2.80%, 93.35 ± 3.21%, and 90.51 ± 7.32%, respectively. The proximal femur diagnostic outcomes for the LP-U-net, V-net, HighResNet, and 3D U-net models were reported as 98.14 ± 0.47 %, 96.89 ± 0.85 %, 96.47 ± 1.54 %, and 89.99 ± 4.91 %, respectively. The findings from the suggested network architecture indicate that HDC and DUC might be effectively combined with other FCNs to get the highest level of performance in the task of semantic segmentation.

Chapter 3

Methodology

3.1 System Model

The suggested system uses the U-Net architecture to conduct brain MRI segmentation, to identify tumor core, and total tumor, and to enhance tumor areas, all of which are interconnected in some way. To achieve this, the U-net model necessitates the construction of a mechanism that accepts three-dimensional images as inputs, then the systematic extraction of required characteristics from those images, followed by segmentation. One of the significant additions to our 3D U-Net is incorporating 3D Tiled Convolution into our model. 3D-TC significantly reduces GPU memory for MRIs such as the one provided in the BraTs dataset. One of the changes in our UNet is that we have added a Dense Upsampling Convolution layer before the $1 \times 1 \times 1$ convolution layer; hence resulting in an upsampled segmentation of the given image with a higher patch size. After the segmentations have been performed it is compared against ground truth labels for better system evaluation.
The proposed system and the 3D-TC approach are network agonistic meaning that the U-Net model and its internal architecture can be replaced with other segmentation architecture that poses a Fully connected network. Hence, this system will allow for improved performance with different FCNs combined with the 3D-TC and DUC layers. To evaluate the tumor ratio, the aforementioned system, which consists of the U-Net model, will be employed to execute segmentations.

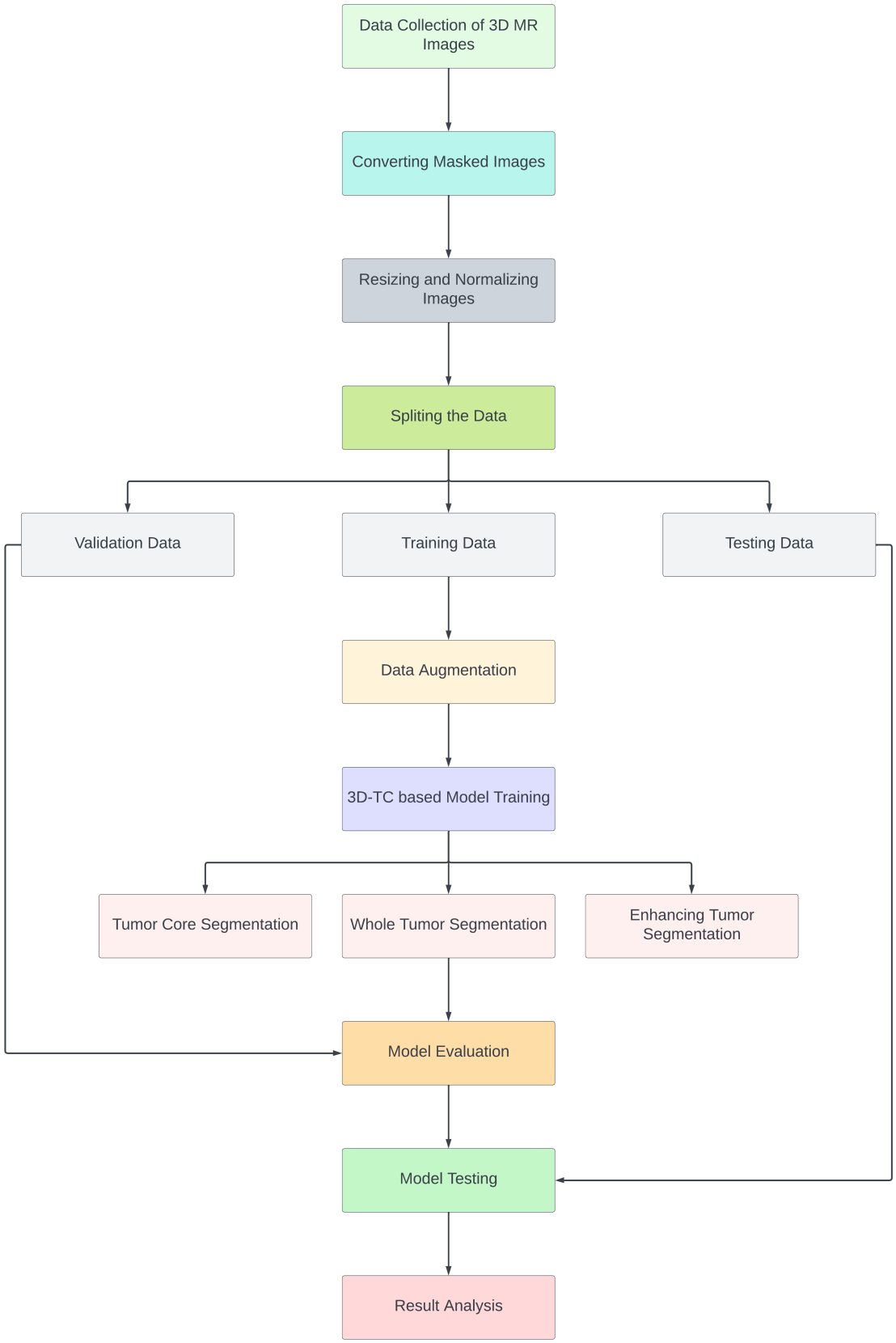

Figure 3.1: The Work Flow of Proposed 3D-TC Brain Image Segmentation Model

3.2 Model Architecture

3.2.1 CNN

A CNN is a kind of artificial neural network in machine learning that takes sensory data as input. CNNs can fulfill a huge variety of purposes [15], such as IP, NLP, and other specialized applications. CNNs are standardized multilayered perceptions made up of multiple layers of artificial neurons. In most cases, a multilayered sense is described using fully connected networks, and all the neuron in every layer is coupled into each neuron in the next layer. As of late, the CNN [26] has been the go-to tool for automatic picture identification and detection. Convolutional neural networks (CNNs) are very productive, and image identification uses surprisingly little computational resources. CNN is a straightforward approach that relies on stacked layers of neurons to achieve detection.

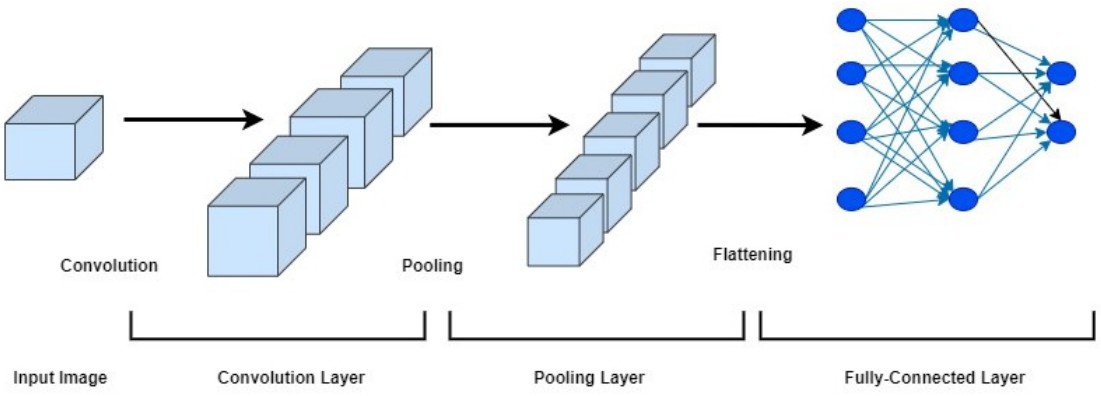

Figure 3.2: CNNs Basic Architecture

3.2.2 FCN

Some higher-class models known as fully convolutional networks can tackle a wide variety of pixel-wise tasks. [10] Instead of using thick layers as CNNs do, as discussed, FCNs employ 1×1 convolutions to simulate the effectiveness of completely 17-connected layers, rather than dense layers as is the case with CNNs. In recent times, many methods have been developed to significantly enhance the accuracy of semantic segmentation, including the export of pre-trained classifier-weights, the combination of multiple layers of representations, and learning the end-to-end on whole images [27]. Primary architecture for FCNs often looks like the diagram below, however, exact details may vary across models.

3.2.3 3D-U-net

Development and application of models vary greatly concerning both the nature of the work and the intended audience. U-net was created specifically for semantic

separating. Its architecture has earned it the name "U-net." In the area of semantic segmentation, it remains one of the most well-liked full-pipe solutions[28] because of its success in a variety of applications. Researchers, O. Ronneberger, first suggested 3D-U-Net in a 2013 publication [29]. Here at 3D-UNet, we provide a variant tailored to Fabian I. et al. work on segmenting brain tumors. With its adaptability, 3D-UNet can be used to solve a wide range of segmentation issues, and it enables the continuous segmentation of 3D volumes with enhanced accuracy and performance. The first half of the U-shaped route, representing a down-sampling process, is known as the encoder path, while the second half, representing an up-sampling operation, is known as the decoder path. Because the actions or routes are joined together, we have regionalized data, which allows for semantic dissection. Here, We give $128 \times 128 \times 128$ input pixels. The typical combination of 3D convolution and 3D max pooling is carried out. Every layer in the architecture consists of two 3x3x3

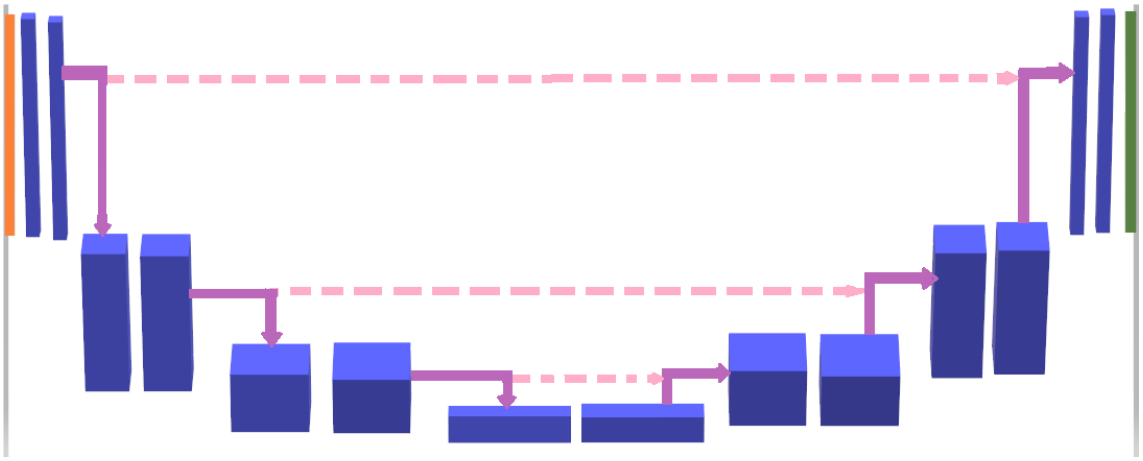

Figure 3.3: Architecture

3D convolutions, a ReLU, a $2 \times 2 \times 2$ 3D max pooling with two stages in each dimension, and a ReLU. In the mixed approach, each layer is constructed from an up-convolution of $2 \times 2 \times 2$ by steps of 2 in each dimension, two $3 \times 3 \times 3$ convolutions, and a ReLU on top of that. The core high-goal items discovered in the research path are transferred to the synthesis path through direct route connections across layers with similar objectives. A $1 \times 1 \times 1$ convolution is used in the last layer further to lower the ratio of result channels to total marks. Each ReLU activation begins with a group standardization process.

3.2.4 Attention Gate

The attention U-Net, first proposed in [30], is an improvement upon the standard U-Net design by including an attention gate mechanism inside the decoder. The attention gate modifies the feature map produced by the encoder before the concatenation in the decoder block. It learns which parts of the encoder's feature map are crucial by examining the feature map of the previous decoder block. The feature map from the encoder is multiplied by the attention gate's estimated weights to get this result. The weight values between 0 and 1 in a neural network represent the amount of attention being paid to each pixel.

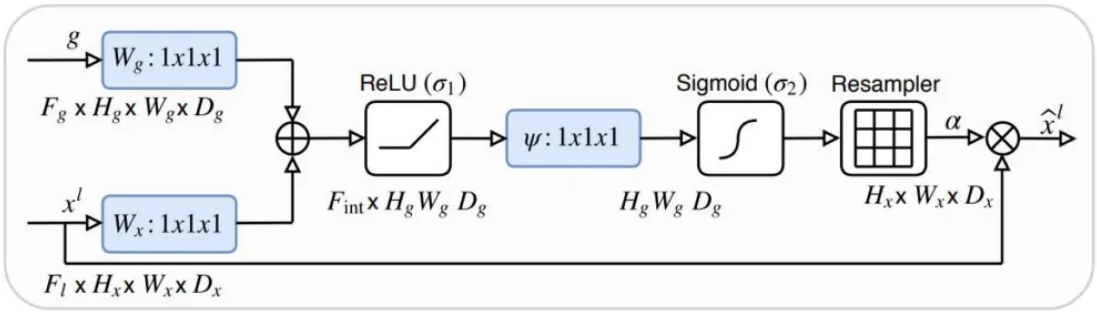

Figure 3.4: The construction of the attention gate, the process involves the multiplication of input characteristics (x^l) with attention weights (α). Initially, the input features (x^l) and the feature map from the appropriate encoder level are subjected to a $1 \times 1 \times 1$ convolution operation, followed by the computation of their sum. Subsequently, a rectified linear unit (ReLU) activation function is used, followed by an extra $1 \times 1 \times 1$ convolution operation.

3.2.5 Residual Connection

Residual U-Net was proposed in [31] which took inspiration from the ResNet model which proposed the residual connections to better mitigate the vanishing gradient problem. When training a deep neural network, the addition of residual connections improves gradient flow. The difference between the traditional convolution block used in U-Net and the residual block used in residual U-Net is depicted in the figure demonstrated below:

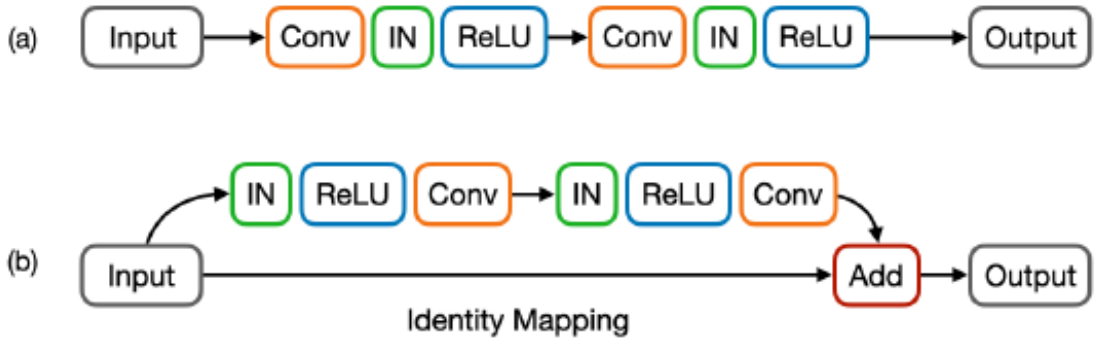

Figure 3.5: Distinctions between the building components of (a) a standard U-Net and (b) a Residual U-Net. An instance normalization operation is represented by the green IN, a Rectified Linear Unit activation by the blue ReLU, and a convolutional layer's $3 \times 3 \times 3$ kernels by the orange Convolution block.

3.3 3D Tiled Convolution

The fundamental elements of 3D tiled convolution consist of low-resolution (LR) convolutions and a periodic down shuffling operator. The design of 3D-TC is guided by two goals that prioritize quick relevance to the supplied data. In the first phase, the 3D-TC method aims to acquire a set of independent kernels inside a single

layer, with dimensions $s_x \times s_y \times s_z$. Here, s_x, s_y, and s_z represent the variables that consider the down-shuffling along with the three axes. In contrast to conventional convolutions, which involve the sharing of convolutional kernels across all neurons within a certain layer, this approach enables the use of separate kernels for individual neurons. In contrast to a simple down-sampling procedure, the whole of the input data is employed without consideration of the down-sampling factors. The use of a three-dimensional tensor decomposition technique, known as 3D-TC, has promise in efficiently reducing data size to conduct non-linear analysis. The anticipated output dimensions of 3D-TC are expected to be $(d \cdot h \cdot w) \cdot k$, where C represents the number of input channels in the input volume and k represents the number of output channels from the LR conv layer. Here, d, h, and w denote the dimensions of the input and output, respectively. Before applying convolutions with a kernel size of $3 \times 3 \times 3$, we use a periodic down-shuffling operator on the input data to generate $C \times (d \times s_x)(h \times s_y)(w \times s_z)$ channels of low-resolution (LR) feature maps.

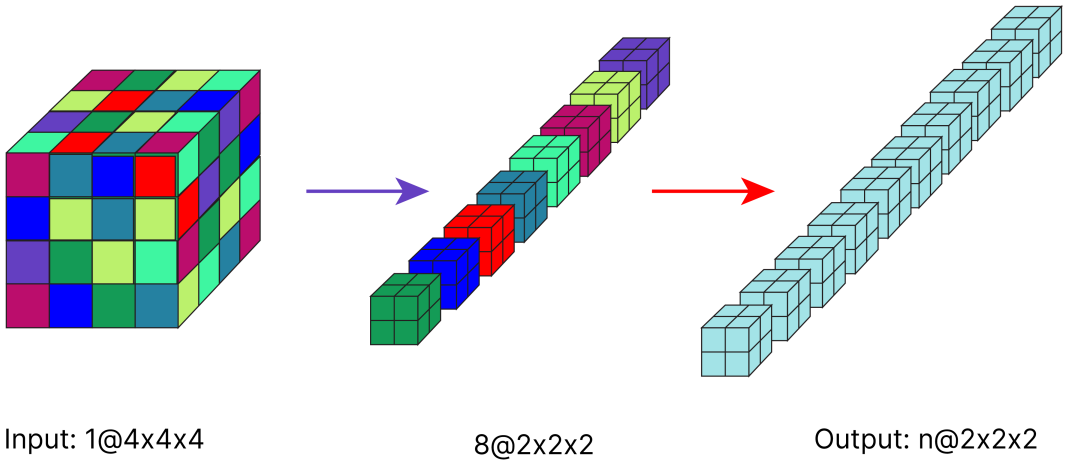

Figure 3.6: 3D Tiled Convolution Schematic View

As demonstrated in Fig. 3.6, the 3D-TC technique comprises a PDS operation followed by a low-resolution (LR) convolution. The periodic down-shuffling process is implemented by using the pixel shuffle approach as described in the work of Shi et al. (2016). In contrast to conventional convolutions, which involve the sharing of kernels among all neurons in a given layer, the 3D tiled convolution approach

distinguishes itself by seeking to learn separate kernels of dimensions $s_x \times s_y \times s_z$ within a single conv layer. Here, s_x, s_y, and s_z represent the down-shuffling factors along the three spatial axes, respectively. Hence, regardless of the down-shuffling factors used on the input data, all data will still be available for the next layer to use; which differs from the traditional down-sampling techniques. Secondly, the most significant benefit of the 3D-TC approach is the reduction of the data size for subsequent processing that needs to be executed by the model. Assuming that the high-resolution input data (I^{HR}) is of size $(d \times s_x)(h \times s_y)(w \times s_z) \times C$, the output of 3D-TC will have a size of $d \cdot h \cdot w \cdot k$, where C is the number of channels in the input data. To obtain the k feature maps of size $(d \times h \times w)$, we did not perform subsequent convolution to the high-resolution image but instead used a periodic down shuffling operation on the input to obtain the $C \times (s_x \times s_y \times s_z)$ channels of low-resolution feature maps. The following equations describe the 3D-TC mathematically.

$$TC(I^{LR}, W_l, b_1) = \phi(W_l * PDS(I^{HR}) + b_l) \tag{3.1}$$

Here, the ϕ refers to a non-linear activation function; W_l, b_l are the weights and bias of last layer; PDS signifies the periodic-down-shuffling operation which is used to rearrange the high-resolution space tensor (T_{HR}) from the shape of $(d \times s_x)(h \times s_y)(w \times s_z) \times C$ into the low-resolution space tensor (T_{LR}) which will be the shape of $(d \cdot h \cdot w) \times (s_x \times s_y \times s_z \times C)$. The operation to get the $T_{LR} = PDS(T_{HR})$ can be represented as the following:

$$\begin{aligned} T_{LR}(x', y', z', c') = T_{HR}(&x' \cdot s_x + \lfloor mod(c', s_x \cdot C)/C \rfloor, \\ &y' \cdot s_y + \lfloor mod(c', s_x s_y \cdot C)/(s_x \cdot C) \rfloor), \\ &z' \cdot s_z + \lfloor c'/(s_x s_y \cdot C) \rfloor, \\ &mod(c', C)) \end{aligned} \tag{3.2}$$

In the given context, the variables x', y', and z' represent the coordinates of the voxels in the low-resolution space. Specifically, x' takes values within the interval $[0, d-1]$, y' takes values within the interval $[0, h-1]$, z' takes values within the interval $[0, w-1]$ and c' takes values within the interval $[0, C \cdot s_x \cdot s_y \cdot s_z - 1]$.

3.4 Dense Upsampling Convolution

For a fully convolutional network that includes the downsample-upsample path, dense prediction can be achieved on input volumes, we require the full resolution at the output. To retrieve the full resolution in the output conventional methods like the bilinear upsampling [32] or trilinear upsampling are not as effective due to the upsampling parameters being not learnable. The deconvolution layer proposed in [33] is an alternative to the conventional techniques but even the deconvolution layer can easily introduce a problem known as "check board artifacts" which is caused due to the uneven overlap between the neighboring receptive fields during the upsampling procedure. Secondly, deconvolutional layers have a high processing overhead. Upsampling is a computationally and resource-intensive process, which may increase training durations and memory use, particularly for big feature maps. In [34] an efficient sub-pixel convolution was introduced which consisted of LR space convolution followed by the periodic up-shuffling operator; which was performed by

the pixel shuffle technique. This was done to retrieve the HR space image from a low-resolution counterpart also known as "super-resolution". The mathematical procedure of the ESPC is described below:

$$I^{HR} = f^n(I^{LR}) = PUS(W_n * f^{n-1}(I^{LR}) + b_n) \tag{3.3}$$

Here, PS rearranges elements of $D \times H \times W \times C * (s_x \times s_y \times s_z)$ tensor to a tensor of shape $s_x D \times s_y H \times s_z W \times C$. Mathematically the periodic up-shuffle operation can be broken down in the following way:

$$PUS(T)_{x,y,z,c} = T_{\lfloor x/s_x \rfloor, \lfloor y/s_y \rfloor, \lfloor z/s_z \rfloor, C \cdot s_z \cdot mod(z,s_z) + C \cdot s_y \cdot mod(y,s_y) + C \cdot s_x \cdot mod(x,s_x) + c} \tag{3.4}$$

The convolution operator W_L thus has shape $n_{L-1} \times s_x * s_y * s_z * C \times k_L \times k_L \times k_L$; where k_L denotes kernel size.
The DUC workflow combines of and $3 \times 3 \times 3$ convolution operation followed by normalization and a ReLU function and then PS is finally applied to get the resulting upsampled volume which can be used for the output prediction. We have demonstrated the block diagram for the DUC below:

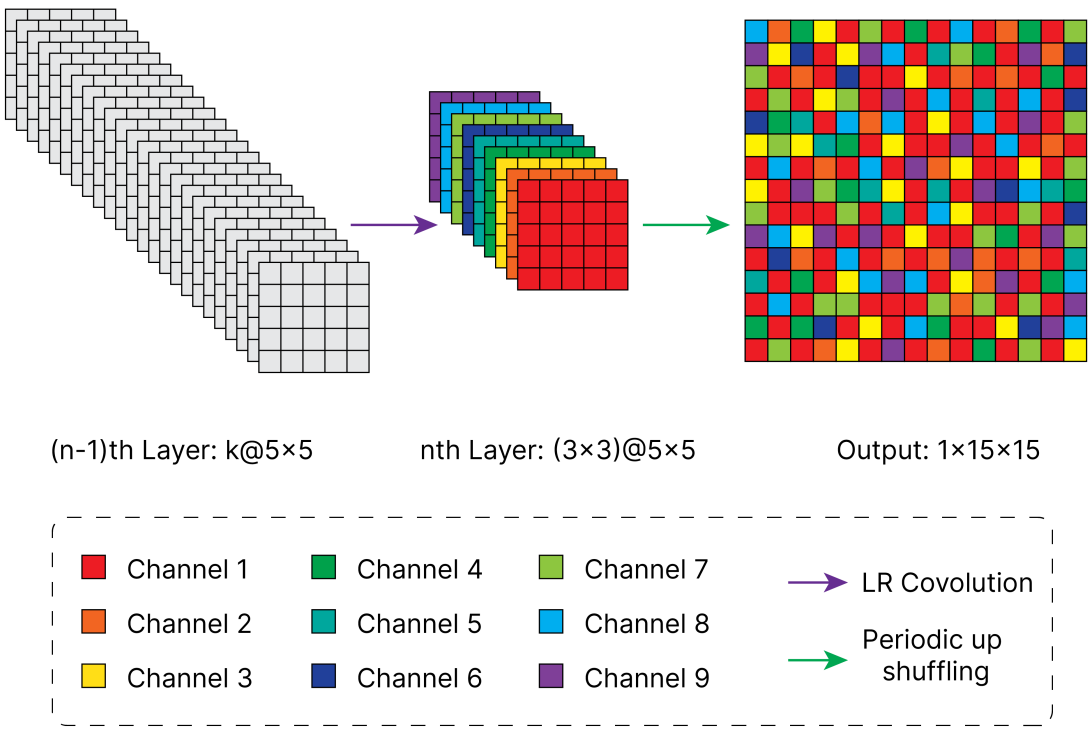

Figure 3.7: Dense Upsampling Convolution Schematic Diagram showcasing 3x upsampling of 5×5 feature maps from $(n-1)^{th}$ layer to 15×15 output

3.5 Deep Supervision

Deep supervision [35] is a technique where loss functions are computed from different decoder levels for better gradient flow. In our experiment, we utilized two more output heads marked by the sigmoid activation present in the second and third

decoder levels in Fig. 2. To calculate the deep supervision loss, we obtained supplementary ground truth labels with dimensions (96, 96, 64) and (64, 64, 32) by the use of closest neighbor interpolation. This was done to align the form of the additional decoder-level outputs. The calculation of the final loss function is determined by the labels gt_i and predictions p_i for three specific output heads. These output heads are labeled as i = 1, representing the last output head, i = 2, representing the output head on the penultimate decoder level, and i = 3, representing the output head before the penultimate decoder level.

$$f_l(gt_1, gt_2, gt_3, p_1, p_2, p_3) = f_l(gt_1, p_1) + \frac{1}{2} f_l(gt_2, p_2) + \frac{1}{4} f_l(gt_3, p_3) \tag{3.5}$$

Chapter 4

Implementation

4.1 Dataset Description

For our purpose we used the BraTS'20 MICCAI dataset [36]–[39]. This dataset went through a lot to get to BraTS'20. Because some of the datasets had been personally annotated by clinical specialists, there was a large discrepancy between the datasets published by this source. In this respect, the BraTS'17-'20 had some overlaps with the BRATS12-13 provided images and annotations. In addition, The Cancer Imaging Archive datasets from BRATS14–16 were thrown out. One reason is that the scan descriptions span both pre and post-operative times; another is that methods that performed well in BRATS12 and 13 were used to annotate the ground truth labels of the datasets. The initial Cancer Imaging Archive glioma collections, which included 262 TCGA-GBM and 199 TCGA-LGG, were reportedly radiologically evaluated and categorized by expert neurologists. Before surgery, 135 glioblastoma multiforme (GBM) and 108 glioblastoma (GBG) images from The Cancer Imaging Archive were manually labeled for different glioma subregions.
Here, NIFTI-formatted ".nii.gz" files store multimodal scans. The dataset contains 369 and 125 well-arranged folders of annotated brain tumor images for training and validation purposes. There are 5 "nii" files in each training image description T1 being native, T1Gd being post-contrast T1 weighted images, T2-weighted, Fluid Attenuated Inversion Recovery volumes being T2-flair and seg. The "seg.nii" and other validation files include all of these. These documents are generated by the training model. The Cancer Imaging Archive is where you can find both the TCGA-GBM and TCGA-LGG collections, as well as data sheets for name mapping and survival statistics. In addition to the ages of the persons whose MRI scans were shown, the file also included information regarding the patients' resection status. The data was compiled using clinical procedures and scanners from about 19 different hospitals. The ".nii" files that contain our data are shown in a few examples below. Images have X, Y, and Z dimensions of 240 by 240 by 155. Each voxel is 1 mm on a side and 1 mm in volume. Training data were manually segmented by anything from one to four raters using the same annotation process. The neuroradiologists checked the labels to make sure they were accurate. According to the TMI articles from 2012–2013 and the most recent summary studies from the same dataset, the GD-enhancing tumor is labeled as ET (label 4), followed by the peritumoral edema (label 2), and lastly the necrotic and non-enhancing tumor core (label 1). Inflated to the same pixel count after skull removal, they are all co-rosters for the same

anatomical templates.

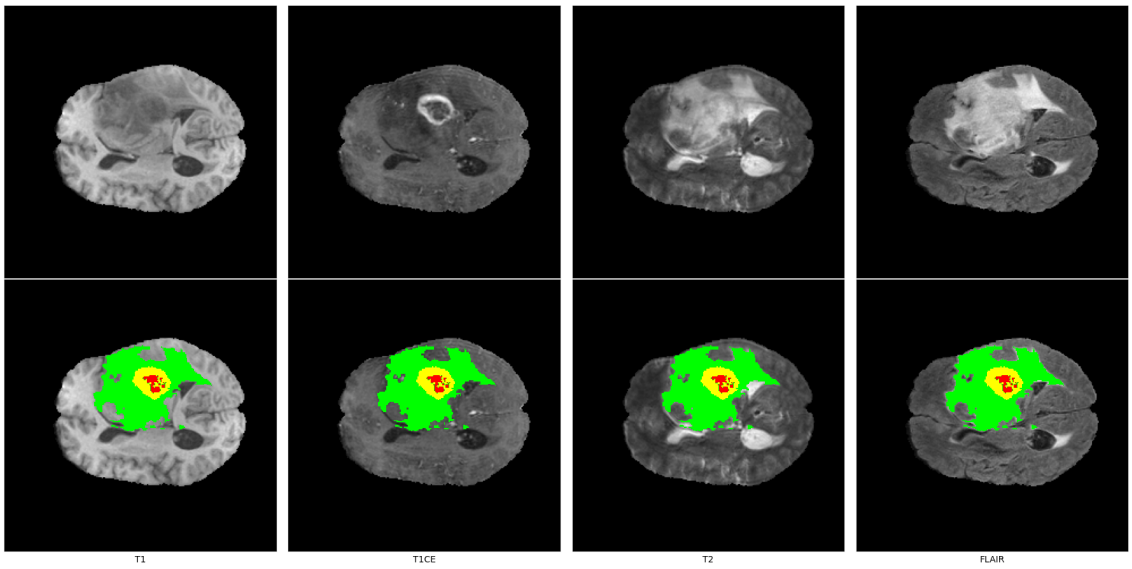

Figure 4.1: BraTS 2020 example of training dataset scan and corresponding annotations of different modalities. Each region is highlighted in a different color: red for the NCR/NET, green for the ED, and yellow for the ET.

4.2 Data Preprocessing

For the image segmentation, we had to pre-process the BraTS'20 MICCAI data and then use it. We received four types of data, or four distinct volumes of the same area, in the dataset. There is t1, which is rather light and simple to notice; t1ce, the t1-weighted image with strong contrast to make it visible; T2-weighted; and T2 flair, all of which aid in bringing out the finer features of the structures. We were able to get images in t1, t1ce, t2 and flair. T1 images have higher contrast and it allows us to see more detail. As the outside of the images has some blank portion which is unnecessary, we removed them to gain a clearer perspective for data training and a more conducive physical environment.

4.2.1 Normalizing and Combining Channels

We had to normalize the intensity value of all 4 given channels flair, t1, t1ce, and t2 so that it could be utilized in the model afterward. For this, we have used MinMaxScaler from the sci-kit learn library. To normalize the 3D images we needed to convert the channels into 1D by reshaping them and reconverting them back to the original dimensions. To normalize there will be a size reduction, resulting in an even more improved version of the raw data that was initially collected. The information from the dataset indicated that the value of the pixels required to be 4, but because it was absent from the original labels, it was changed to 3. We saved the numpy arrays (npy) from each folder and merged the t1, t2, and t1ce, flair images into a single-multi channel volume to utilize as our 3D input for the suggested model. The technique of merging the volumes should result in a greater improvement in the data because employing individual volumes should have less of

an impact. As a result, our dataset was completely prepared to be used for the training and validation of our proposed model. Because fixed area cropping would not provide a lossless solution in this case, we allowed our data to be cropped using a bounding box approach. By using this technique, there were some parts of the images that were missing and it limited the area of the outside box also it made possible for us to eliminate those parts from the picture. As a result, we can say that more pre-processing of the data is required to preserve all of the sections of the data also the exception of the region of the data denoted by the label 0 and the blank area of the data. This was done to ensure that we would not lose any of the data obtained from any of the slices captured. As a direct consequence of this, one of our objectives for the post-proposed pre-processing of the data is to acquire a more precise segmentation validation.

4.2.2 Data Augmentation

Data augmentation is a method that may be efficiently used to expand the training dataset and mitigate the issue of overfitting, which often arises during the training process when models struggle to generalize when exposed to validation data. In our experimentation, we used the following data augmentations:

1. **Random biased crop:** From the input volume of (4, 244, 244, 155) in dimension, a patch of dimensions (4, 192, 192, 128) was randomly cropped. This specific dimension of patch size was chosen due to the higher encoder level of the deeper u-net that was used for evaluation. Along with that, with a probability of 0.5 the patch used via the random crop ensured that there was a significant amount of foreground voxels present in the cropped region.

2. **Intensity scaling:** Random intensity scaling was incorporated to enhance the contrast of the input; here the intensity was scaled with a factor of 0.1 and with a probability of 1.0.

3. **Flips:** With a probability of 0.5, for all of the axes (x, y, z) independently, the input volume was flipped along the specified axis.

4. **Gaussian Noise:** To improve the precision of feature extraction in the training phase, Gaussian noise was introduced to the LGG dataset with a probability of 0.15. To increase the input volume, Gaussian noise is introduced to each voxel. The noise is characterized by a mean of zero and a standard deviation that is randomly selected from the range of values between 0 and 0.1.

5. **Random zoom:** The input volume was scaled to its original size using closest neighbor interpolation, and a random value was selected uniformly from (1.0, 1.2) with a probability of 0.15. The same interpolation was used for the ground truth.

6. **Brightness:** A uniformly selected random number between (0.5, 1) was multiplied by the voxels in the input volume with a probability of 0.15.

4.3 Model Implementation

4.3.1 3DTC U-Net

The implemented 3D-TC U-Net architecture can be distinguished into two distinct parts i.e. the encoder path or downsampling path and the decoder or upsampling path. The distinction in our 3D-TC U-Net model is the addition of a 3D Tiled Convolution Block before the contracting path; which groups the feature maps of the high-resolution input into distinct channels hence in essence down-shuffling the input rather than downsampling it using conventional algorithms. The encoder exhibits a modular configuration comprising convolutional blocks. In Fig. 4.2, it can be observed that each block is comprised of two smaller blocks of transformations, specifically denoted by the light purple and light gray colors. The initial smaller block employs a conv layer with $3 \times 3 \times 3$ kernels and a stride of $2 \times 2 \times 2$ to decrease the spatial dimensions of the input feature map by a factor of two. Subsequently, instance normalization and ReLU activation are applied in the light purple block. The subsequent convolution block has a similar setup except it has a stride $1 \times 1 \times 1$ (light gray) which transforms the feature maps from the previous layer. Due to the use of $192 \times 192 \times 128$ patch size for our model, the feature map is transformed to the size of $3 \times 3 \times 2$ at the bottleneck of the U-Net model. The decoder modular structure enhances the spatial dimensions by shrinking the feature map used by the encoder path. Three smaller blocks make up the main block in the decoder. The first block uses trilinear upsampling to make the feature map's spatial dimensions four times larger. Next, the encoder's feature map from the same spatial level is concatenated to the upsampled feature map, and both maps are changed by two identical blocks including a convolutional layer with kernels of $3 \times 3 \times 3$ and stride $1 \times 1 \times 1$, instance normalization, and ReLU activation (light gray). In addition to the decoder block, we use deep supervision to compute the loss functions from the lower level of the decoder (the blue-colored sigmoid block depicts the used decoder levels in Fig. 4.2).

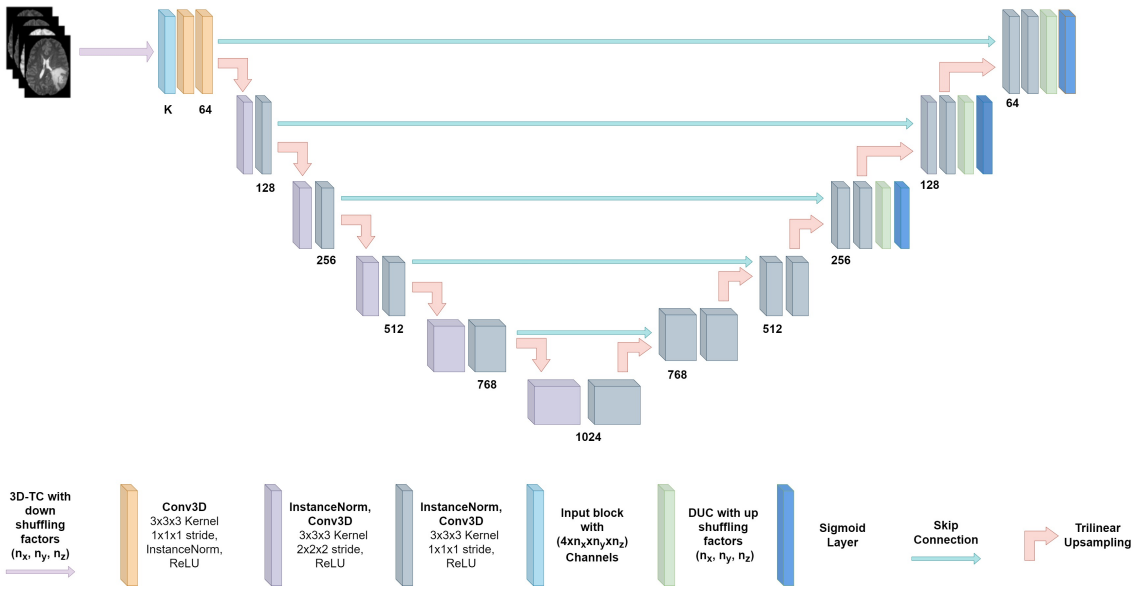

Figure 4.2: Implemented 3D-TC Unet Model Architecture

4.3.2 3DTC R2AU-Net

Inspired by the 2D implementation of the R2AU-Net we adopted the 3D Tiled Convolution with a 3D variant of the R2AU-Net. Similar to the conventional U-Net architecture this U-Net variant also consists of an encoder path and decoder path. The encoding path of the 3D R2AU-Net can be broken into 4 distinct steps. To enhance the model's ability to integrate contextual information, each stage has a recurrent residual convolutional unit. This unit consists of two convolutions with a size of $3x3$ and introduces recurrent connections to both convolutional layers. In each recursive residual convolutional block, the feature maps undergo a reduction in size by half and an increase in number by a factor of two. The design that has been implemented involves the expansion of the RCL block to two consecutive time steps, which are represented by T=2. The RCL architecture comprises a solitary convolutional layer and two recurrent convolutional layers that are partitioned into subsequences. Within the decoder route, the process of upsampling is executed via the use of R2CL blocks, shown by their light purple color. Before concatenating the output R2CL block and feature map from the encoder path; they are passed through an attention gate which readjusts the output features. By producing a gating signal, the attention module adjusts how much distant features matter. Without clipping ROI areas across networks, attention gates progressively decrease feature responses unrelated to background regions. To the final decoder output, DUC is applied to get the segmentation output which is then fed through the sigmoid layer to receive the required prediction probabilities.

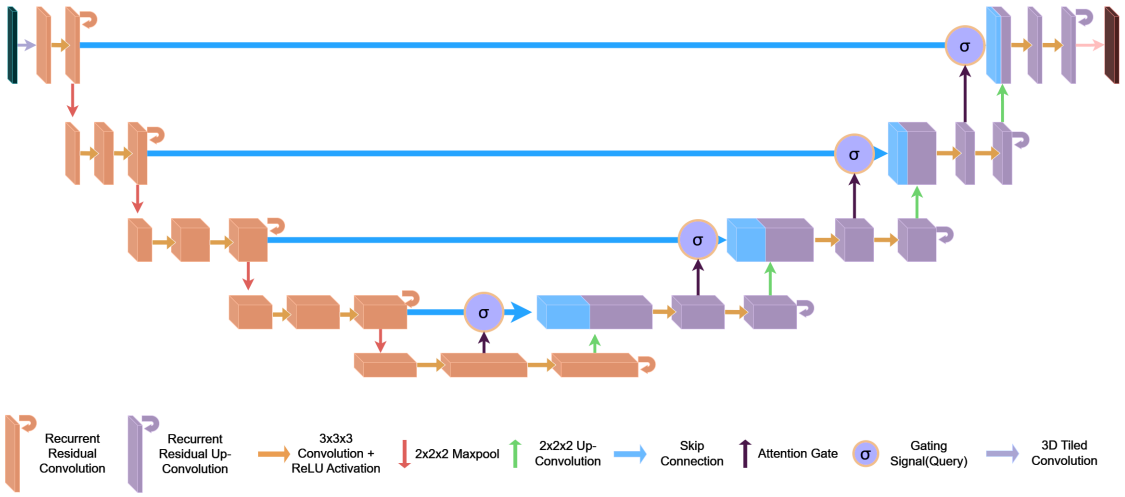

Figure 4.3: Implemented 3D-TC R2AU-Net Model Architecture

4.3.3 3DTC-DUC U-Net

In our experiment, we investigated the effect of replacing the trilinear upsampling done in our 3DTC U-Net with the Dense Upsampling Convolution (DUC) block. In this modified U-Net we needed to change the number of feature maps generated from each of the encoder levels. Instead of having 1024 feature maps the modified U-Net has 384 feature maps of the dimensions $3 \times 3 \times 2$ in its bottleneck. Although we reduced the feature maps we maintained the number encoder-decoder level in

our U-Net; hence the spatial dimensions are the same as the previous U-Net. This change was introduced to reduce the model parameters; without this change, the model would have 264M parameters due to the dense upsampling convolution having to generate $1024 \times (2 \times 2 \times 2)$ features maps before passing it through the periodic up-shuffling operator which will upsample spatial dimensions whilst reducing the number of feature maps. Due to the number of feature maps being 384 at the bottleneck, the model had 110M parameters which is a significant reduction. It should be noted the DUC block applied before each of the sigmoid layers consists of a convolution with a kernel size of $1 \times 1 \times 1$(denoted by the light green block in Fig. 4.4). This model also utilized the deep supervision technique to compute the loss function at the training stage.

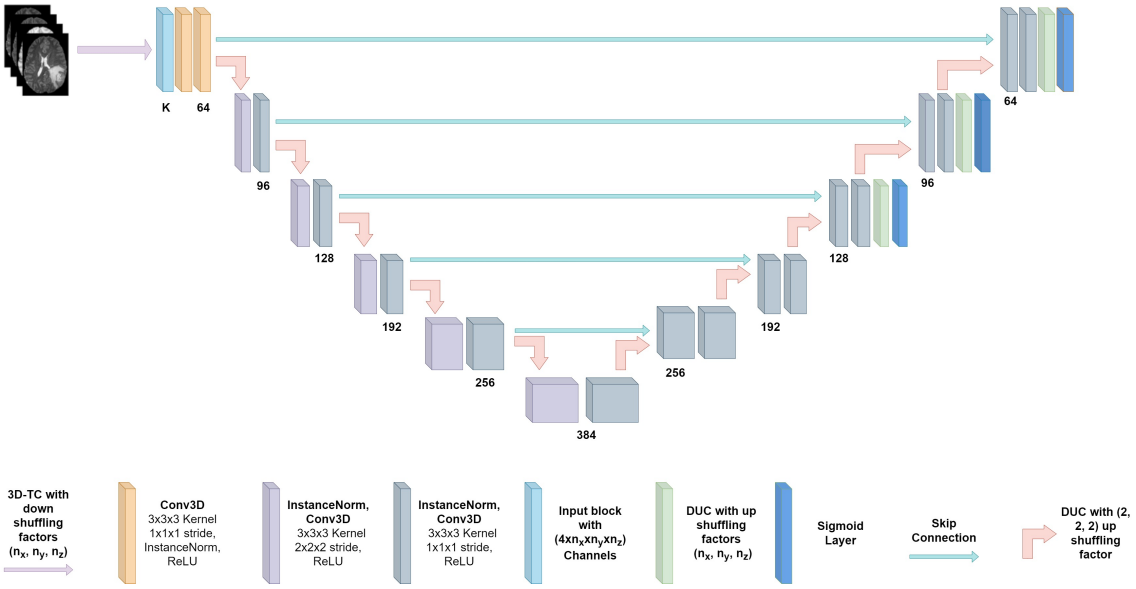

Figure 4.4: Implemented 3D-TC-DUC U-Net Model Architecture

4.4 Evaluation Metrics

4.4.1 Dice-coefficient (F1 Score)

In semantic segmentation, the dice coefficient is used to evaluate the performance of a given model. Another term for it is the F1 score. The model's predicted mask is compared pixel-by-pixel with the mask image to determine their degree of resemblance. To calculate it, divide the overall area of the two images by the total area of the overlap between the two image's segmentation. To do the math, we need to know how much overlap there is between the raw image and the mask image, as well as the total number of pixels in both images [40].

$$DSC = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{4.1}$$

4.4.2 Hausdorff Distance

The Hausdorff distance, also known as the Hausdorff metric or Pompeii-Hausdorff distance, is a mathematical concept used to quantify the separation between two

subsets inside a metric space. The process converts a collection of non-empty compact subsets of a metric space into a metric space of its own. In the context of set theory, it may be said that two sets are near concerning the Hausdorff distance if and only if every individual point included within either set is near at least one point inside the other set. The Hausdorff distance refers to the maximum distance that an adversary must go while selecting a point from one of two sets and then transitioning to the other set. In essence, the quantity being referred to is the supremum of all distances between a point in one set and the nearest point in the other set.

$$HD(P,T) = \max\left(\max_{p\in P}\min_{t\in T} d(p,t), \max_{t\in T}\min_{p\in P} d(t,p)\right) \tag{4.2}$$

4.4.3 Sensitivity

The ability of a model to effectively detect positive instances or cases is quantified by sensitivity, which is also referred to as the true positive rate or recall. The data presented indicates the proportion of cases that are accurately identified as positive and correctly classified as such by the model.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.3}$$

4.4.4 Specificity

Specificity is a metric for evaluating a model's efficacy in properly identifying negative instances. It's a measure of how well the model does at identifying genuine negative situations. The degree to which a model is specific in distinguishing instances that do not hold the condition or event of interest reveals how well the model avoids false alarms or false positives.

$$Specificity = \frac{TN}{TN + FP} \tag{4.4}$$

The equations described above include the variables FP, TP, FN, and TN, which correspond to the ideas of false positive, true positive, false negative, and true negative, respectively. Furthermore, the use of the Hausdorff distance is employed. To ascertain the geographic discrepancy between the areas projected by the model and the regions recognized as ground truth. To address the problem of imprecise forecasts, the Hausdorff95 technique is used as a substitute for the maximum operation. This technique entails using the 95^{th} percentile instead. The symbols "p" and "t" are used as representations for the prediction "P" and the ground truth "T," correspondingly. The symbols d(p, t) and d(t, p) are used to represent the Euclidean distance, which is a mathematical measure that measures the shortest distance between two points, p and t, along a straight line.

4.4.5 Dice Loss

Traditionally, the average per-pixel loss has been calculated discretely using the binary cross-entropy as a loss function when we don't have information about the properties of the surrounding pixels or whether or not they belong to the class we're interested in. This means that binary cross-entropy measures the minimum

loss achievable while disregarding any surrounding information. Therefore, it would be inefficient to use binary cross-entropy to track the prediction loss for semantic segmentation. To prevent this from happening, we employed dice loss, which was derived from the Dice Coefficient [41]. Because the dice-coefficient uses the overlap between the predicted and ground truth pixels to determine similarities, the non-overlapped area represents our model's failure. The probability of losing a roll of dice may be calculated as follows:

$$DiceLoss = 1 - DiceCoefficient \tag{4.5}$$

4.5 Experimental Setup

We utilized a PC equipped with a 3.6 GHz Intel(R) i7 CPU and a 32 GB RAM NVIDIA RTX 3080 Ti graphics card to train the data. We use k=64 as the experimentally determined number of 3D-TC output feature maps. Earlier, we indicated that the resized data we gave included 192x192x128 voxels and 4 different channels. To foretell how well the model will suit the predicted output, we employed 5-fold-cross-validation which means in each of the folds there were 295 training and 74 validation samples. The models were validated after every 10 epochs and an experiment for each of the models was run for 150 epochs. The models mentioned as well as the 3DTC were implemented using the PyTorch framework. The 3DTC U-Net model in our experimentation had 177M parameters due to its higher encoder depth and 3DTC R2AU-Net had 64M parameters. For all of the model experimentation, we used the Adam optimizer with an initial learning rate of 1e-4 and weight decay of 1e-5.

Chapter 5

Result and Analysis

5.1 Training Result

In this section, we will be discussing the calculated evaluation metrics during the training of the various implemented models mentioned in Section 4.3. We evaluate the training loss, accuracy, and dice score gain of individual tumor sub-regions. The 3DTC U-Net model is the only which was trained using a 5-fold cross-validation approach; hence each model generated will be individually represented in this section. It should be worth noting that each of the models went through validation after 10 epochs which is presented in the figures down below. This was done to save the experimentation time.

5.1.1 5-Fold 3DTC U-Net Model

From the Fig. 5.1 we can inspect the initial train loss, and validation loss on the first epoch which was around 74% and 35% respectively for Fold-1. After the 150 epochs completion of the training, the loss was reduced to 12.59% and 12.37%; additionally, we received around 84.5%, 84.1%, 91.2%, and 78.6% mean dice, TC dice, WT dice, ET dice respectively. In our experiment, we reached over 99% accuracy for both the training and validation phases after the 10 epoch.

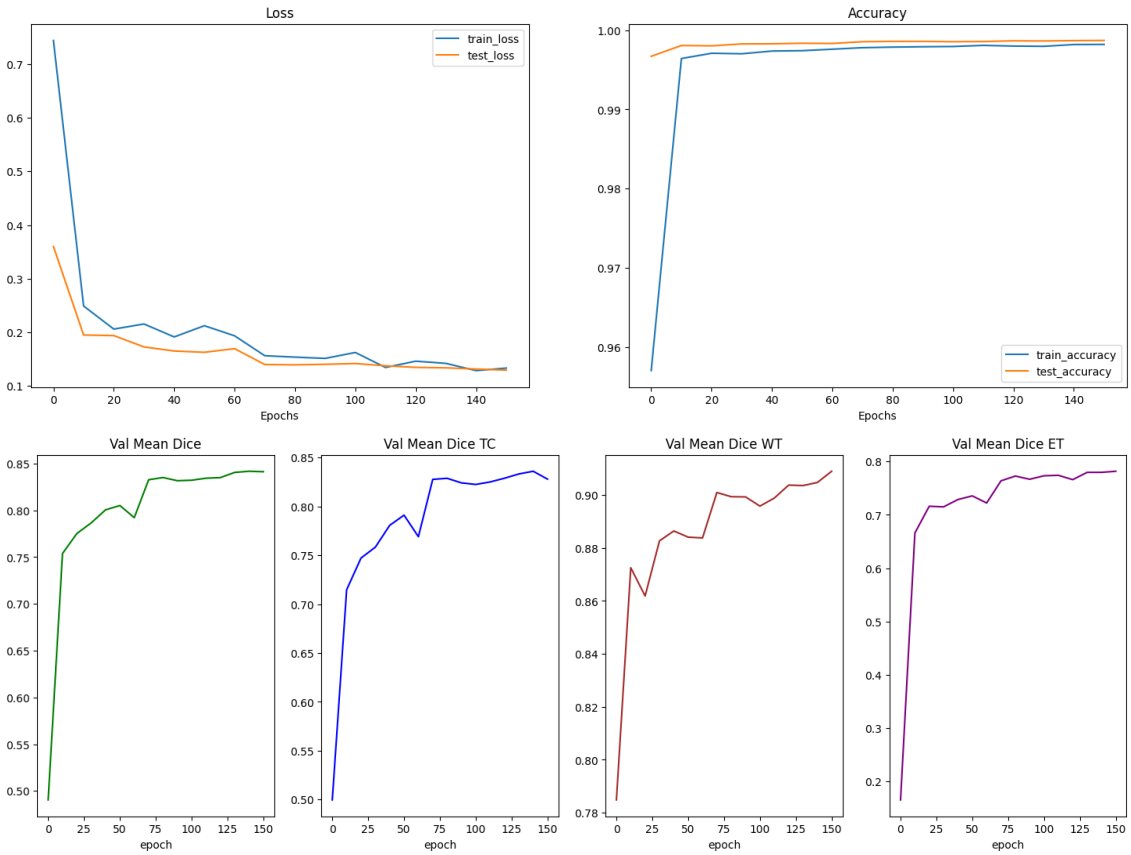

Figure 5.1: Fold-1 Accuracy, Loss and Dice curves for the 3DTC U-Net Model

From the Fig. 5.2 we can inspect the initial train loss, and validation loss on the first epoch which was around 70% and 30% respectively for Fold-2. After the 150 epochs completion of the training, the loss was reduced to 14.13% and 13.71%; additionally, we received around 82%, 81.9%, 89.3%, and 74.12% mean dice, TC dice, WT dice, and ET dice respectively. In our experiment, we reached over 99% accuracy for both the training and validation phases after the 10 epoch.

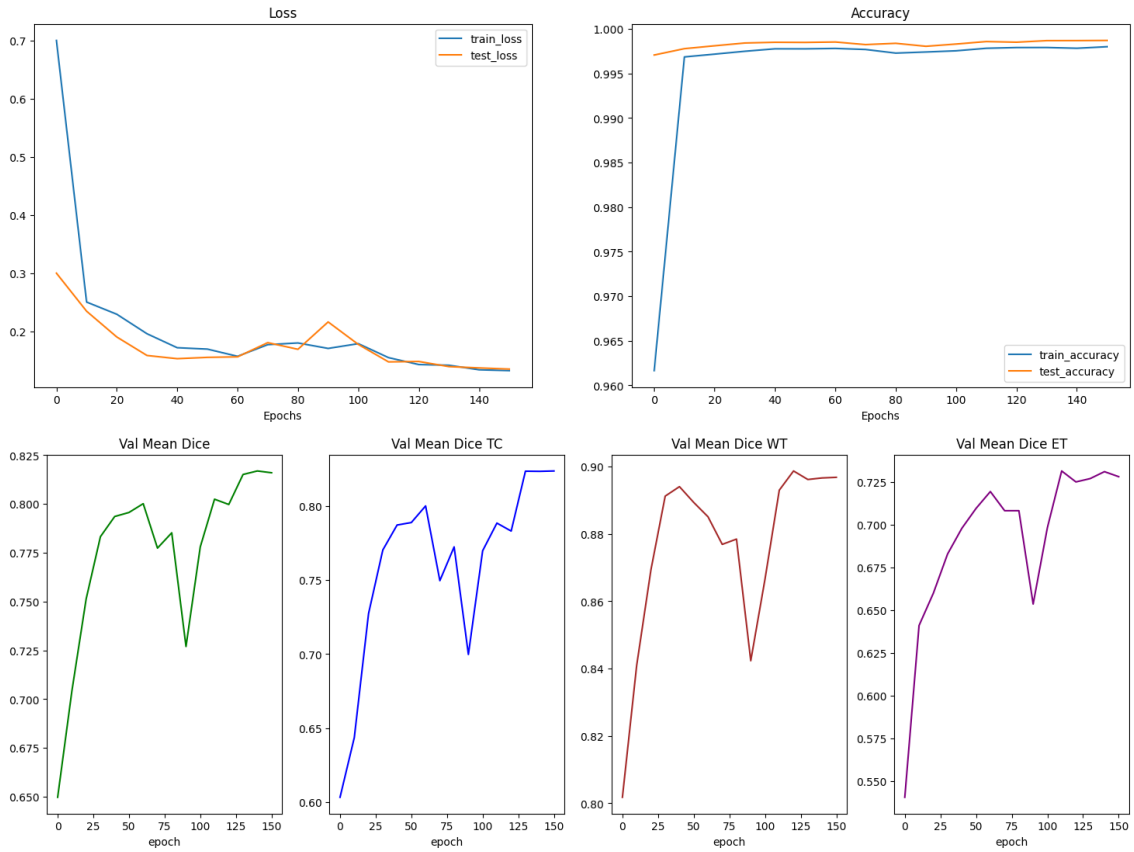

Figure 5.2: Fold-2 Accuracy, Loss and Dice curves for the 3DTC U-Net Model

From the Fig. 5.3 we can inspect the initial train loss, and validation loss on the first epoch which was around 70% and 30% respectively. After the 150 epochs completion of the training, the loss was reduced to 12.29% and 12.17%; additionally, we received around 84.46%, 84%, 91%, and 76.67% mean dice, TC dice, WT dice, ET dice respectively. In our experiment, we reached over 99% accuracy for both the training and validation phases after the 10 epoch.

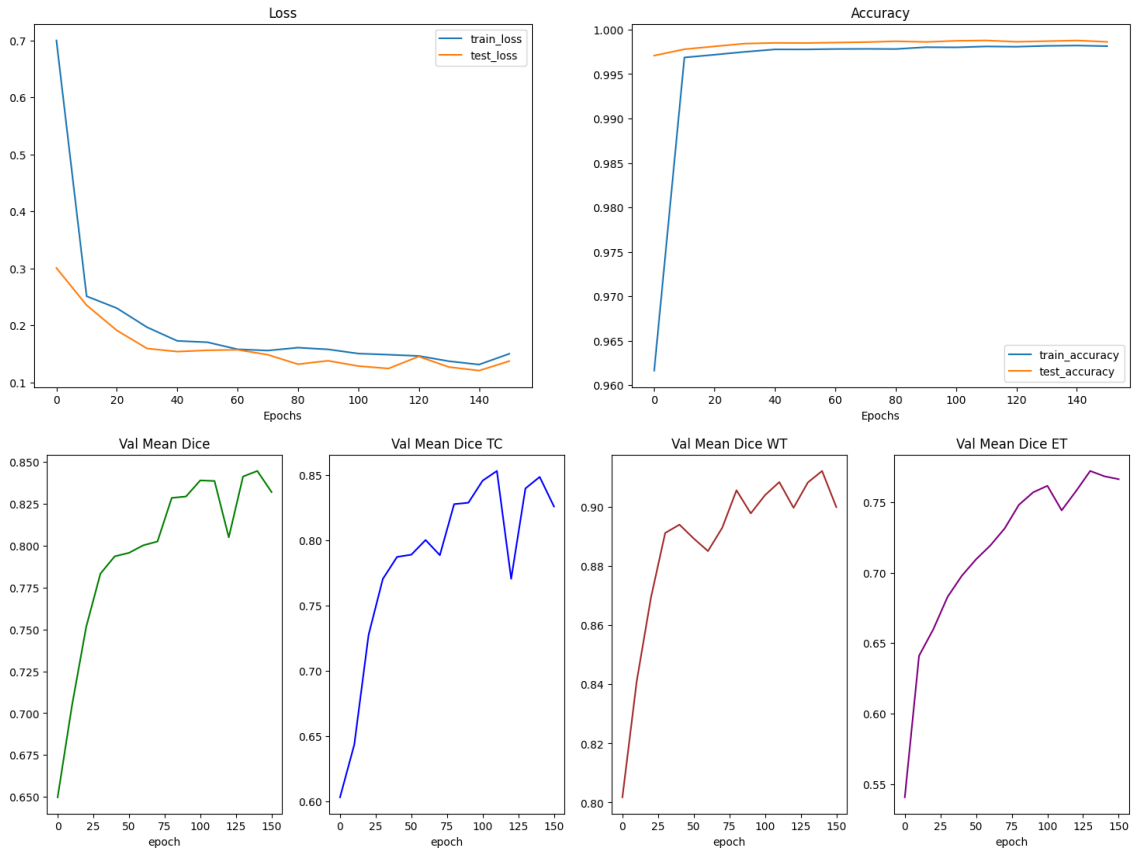

Figure 5.3: Fold-3 Accuracy, Loss and Dice curves for the 3DTC U-Net Model

From the Fig. 5.4 we can inspect the initial train loss, and validation loss on the first epoch which was around 72% and 36% respectively. After the 150 epochs completion of the training, the loss was reduced to 12.37% and 12.29%; additionally, we received around 86.34%, 86.88%, 91%, and 81% mean dice, TC dice, WT dice, ET dice respectively. In our experiment, we reached over 99% accuracy for both the training and validation phases after the 10 epoch. From the Fig. 5.5 we can inspect the initial train loss, and validation loss on the first epoch which was around 73% and 32% respectively. After the 150 epochs completion of the training, the loss was reduced to 13.34% and 11.56%; additionally, we received around 84.12%, 84%, 90.25%, and 79% mean dice, TC dice, WT dice, ET dice respectively. In our experiment, we reached over 99% accuracy for both the training and validation phases after the 10 epoch.

5.1.2 3DTC R2AU-Net

For our 3DTC R2AU-Net model, we have trained it using the Fold-2 of the training dataset. Hence 75% of the training dataset examples were included in the training and 25% was included for the validation of the model. It should be worth noting that due to the higher model complexity of the R2CL block present in this model, the training time was increased. In Fig. 4.9 we have showcased the train loss, accuracy and validation loss, and accuracy curves for the 150 epochs the model was trained for. The dice curves of the R2AU-Net demonstrate the efficacy of the proposed model on different sub-regions of the brain tumor. We attained 80.26% mean dice,

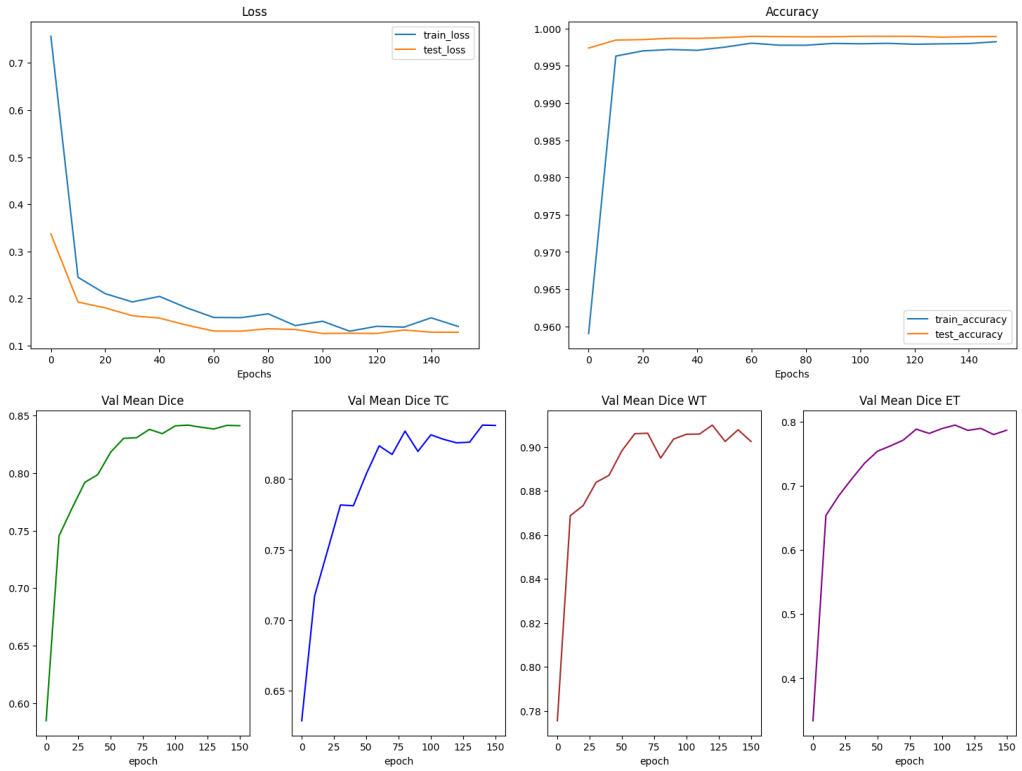

Figure 5.4: Fold-4 Accuracy, Loss and Dice curves for the 3DTC U-Net Model

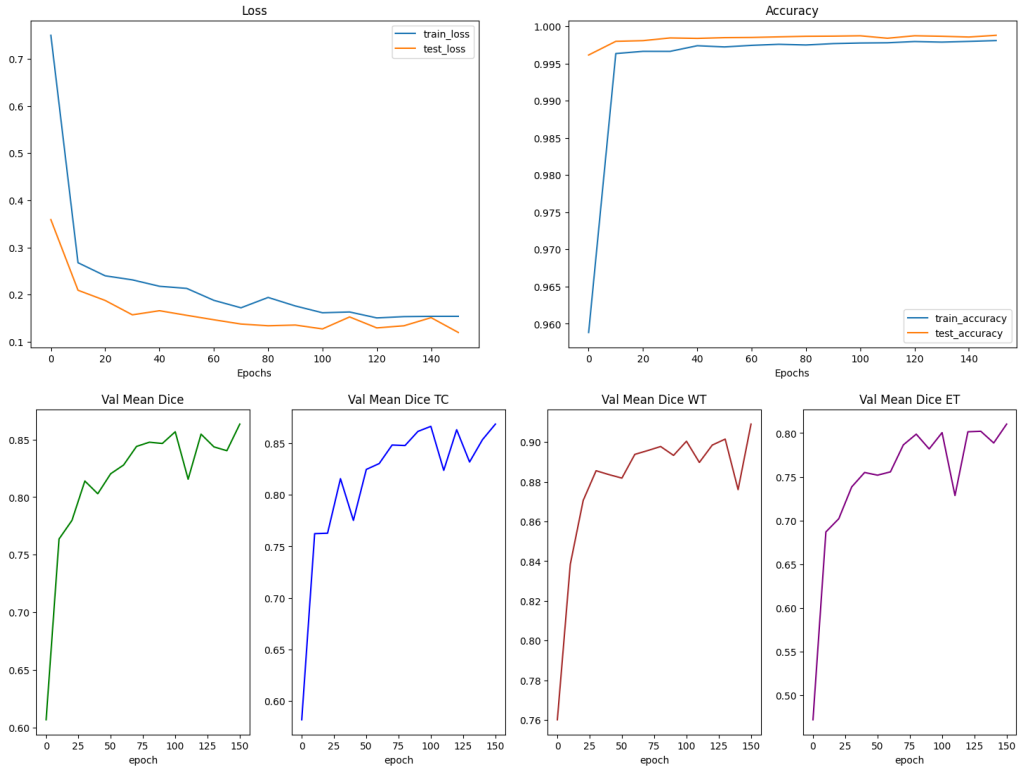

Figure 5.5: Fold-5 Accuracy, Loss and Dice curves for the 3DTC U-Net Model

80.68% TC dice, 88.98% WT dice, and 73.18% ET dice after 150 epochs of training of the model. Although we received over 99% accuracy using this model after 20 epochs; we were not able to exceed the performance of 3DTC U-Net in dice score.

5.1.3 3DTC DUC-U-Net

For the 3DTC DUC U-Net which replaced the decoder path with a dense upsampling convolution block, we trained it on the Fold-2 of the training dataset of the BraTS 20 dataset to gauge the performance impact and further memory consumption benefit. The model produced similar results to the 3DTC U-Net which is prominent by looking at Fig. 4.10. This led us to believe that deep supervision helped reduce the loss effectively hence maintaining significant results on the toughest fold of the dataset. Additionally, the lower number of feature maps produced in the encoder path and DUC incorporation in the decoder path helped reduce the memory consumption by 2 times; where we only required 5.8GB memory for this model even though the patch size was $192 \times 192 \times 128$. For this model, we attained 82.20% mean dice, 81.68% TC dice, 89.28% WT dice, and 74.18% ET dice scores. The final training loss was 13.07% and the validation loss was 12.68% after 150 epochs.

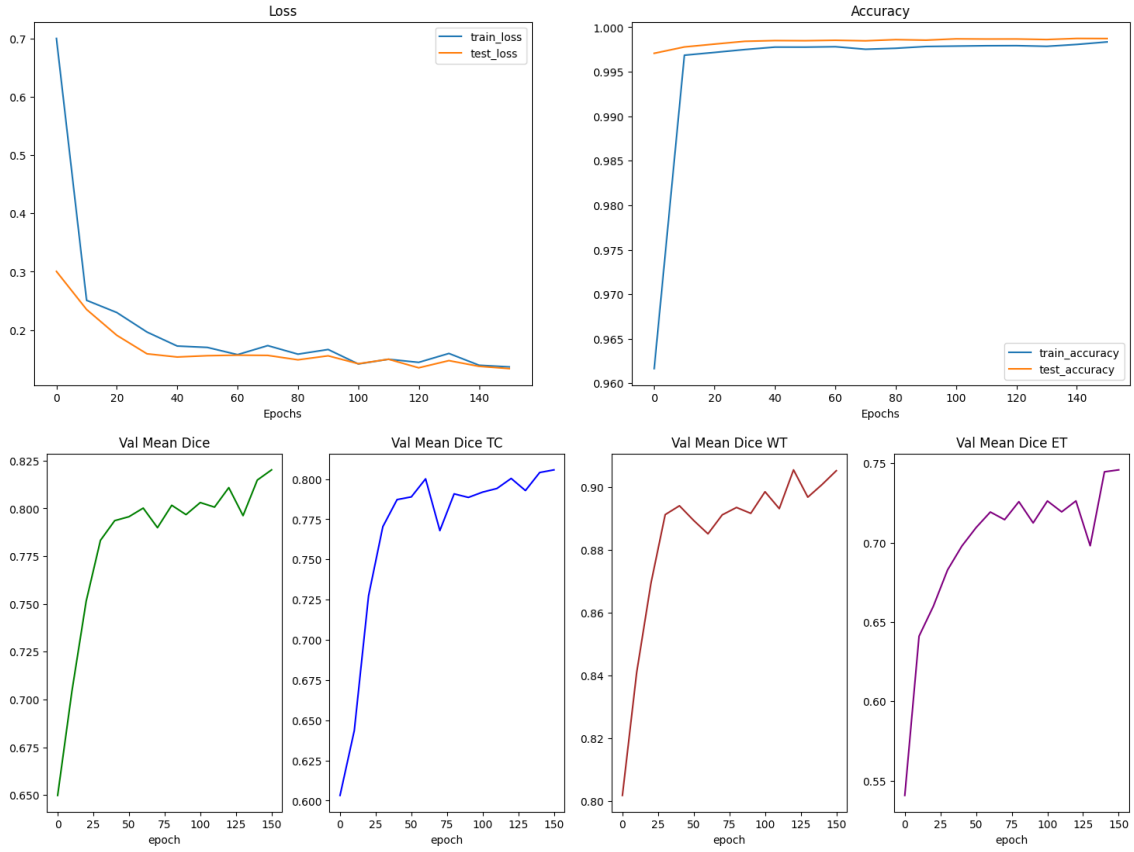

Figure 5.6: Fold-1 Accuracy, Loss, and Dice curves for the 3DTC DUC-U-Net Model

5.2 Effects of Shuffling Factor

In our experiment, we also examined the impact of various shuffling factors that can be employed in the 3D tiled convolution. In this study, we investigated the potential impact on memory consumption and evaluation metric when altering the shuffling factors. To conduct our experiment, we utilized the 3DTC U-Net model, which had demonstrated superior performance in a previous experiment. The dimensions of the input patch were set to $192 \times 192 \times 128$ for all the trained models. The comparisons were conducted using fold 5 of the dataset. Table 4 presents the outcomes obtained from various shuffle factors. Based on the data presented in the table, it can be observed that the optimal outcome was achieved when employing the shuffling factors of (2, 2, 2), primarily due to the increased inclusion of patches. Furthermore, it is noteworthy that even when utilizing the most influential shuffling factors of (6, 6, 2), the segmentation accuracy for all tumor subregions remains below one millimeter. Furthermore, it is important to note that in the case of shuffling factors (4, 4, 2) and (6, 6, 2), we had to decrease the encoder level by one. This adjustment was necessary because the bottleneck would result in a dimension of $1 \times 1 \times 2$, which cannot be effectively upsampled using the decoder path. It is important to mention that a substantial decrease in memory usage was observed when transitioning from a shuffling factor of (2, 2, 2) to (4, 4, 2). Specifically, the memory consumption was reduced from 9.4GB, which was required by the original model and shuffling factors, to a mere 4.8 GB. Fig. 5.8 displays the segmentations that were predicted using different shuffling factors.

Table 5.1: Result from different shuffling factors on the 3DTC U-Net

Shuffle Factor	Dice Score (↑)			Mean Dice
	WT	TC	ET	
3DTC U-Net-(S-2,2,2)	0.9090	0.8688	0.8102	0.8634
3DTC U-Net-(S-3,3,2)	0.8993	0.8616	0.7811	0.8464
3DTC U-Net-(S-4,4,2)	0.8978	0.8569	0.7772	0.8454
3DTC U-Net-(S-6,6,2)	0.8836	0.8332	0.7235	0.8149

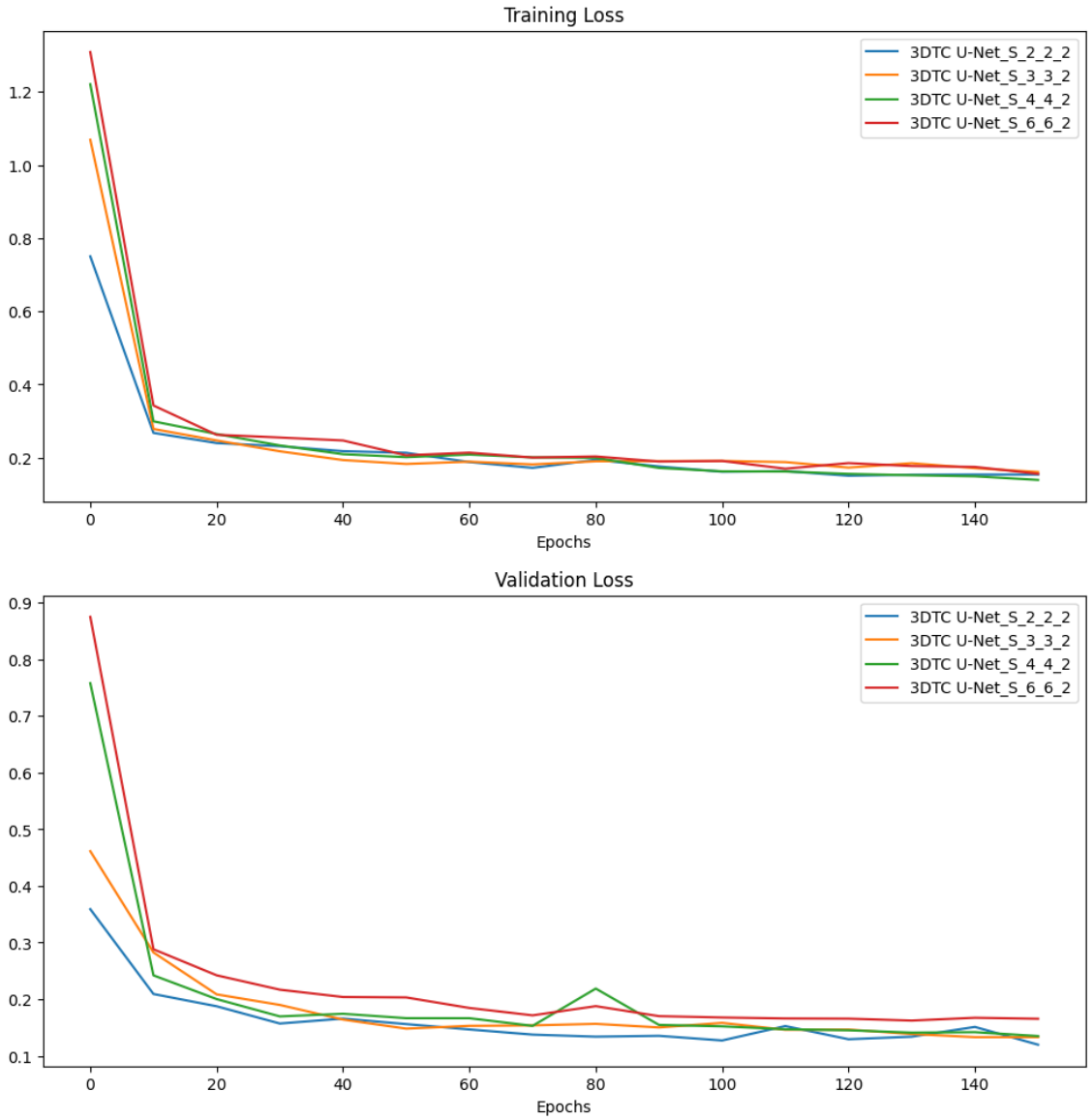

Figure 5.7: Comparison of learning curves using different shuffle factors. The "3DTC U-Net_S_X_Y_Z" denotes the results obtained from the 3DTC U-net with shuffling factors of (s_x, s_y, s_z).

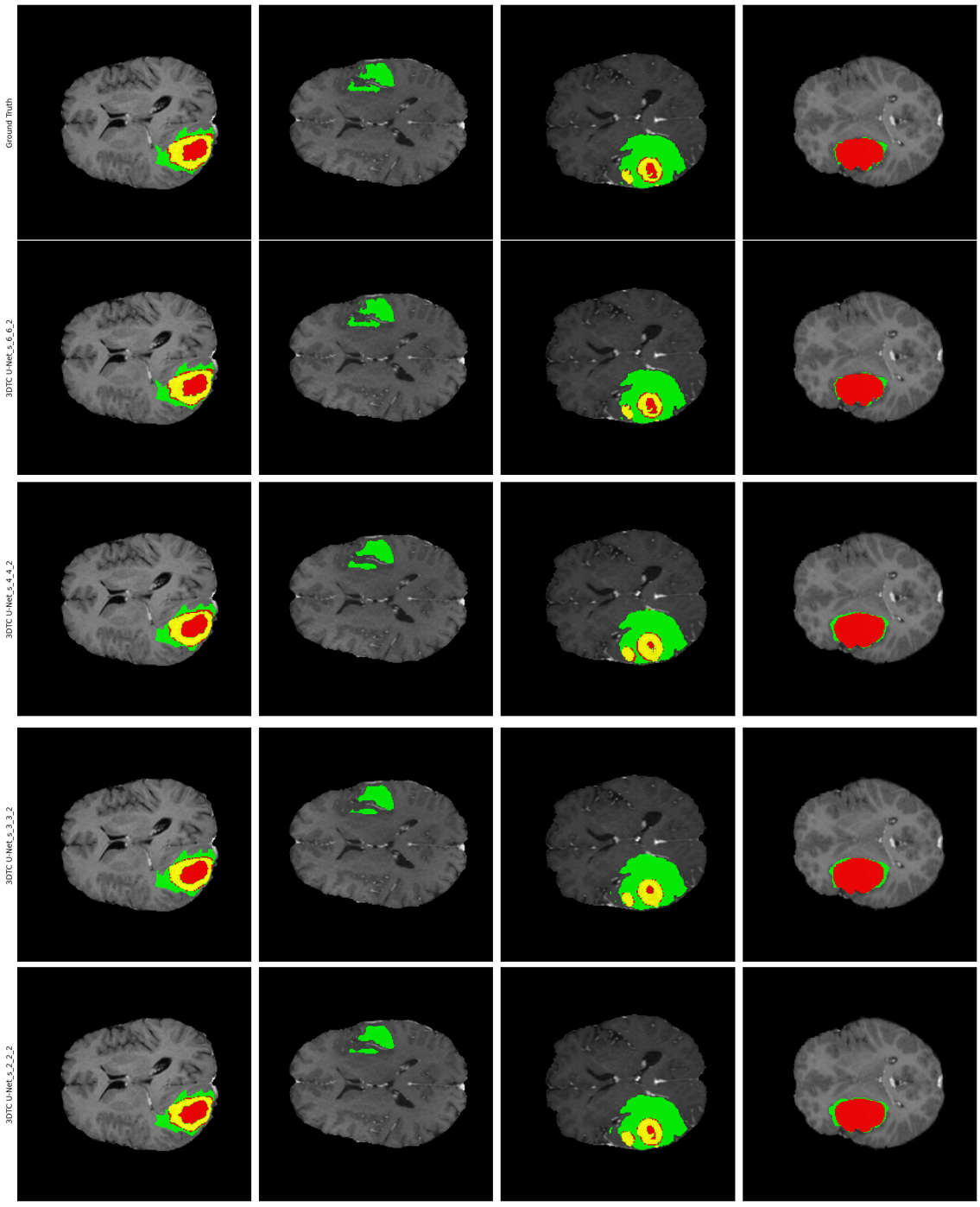

Figure 5.8: Qualitative comparison of the segmentation results from different shuffling factors on 3DTC U-Net model. The first row denotes the ground truth and the following rows are the predicted mask from (6, 6, 2), (4, 4, 2), (3, 3, 2), and (2, 2, 2) shuffling factors respectively.

5.3 Experimental Results

In this section, we evaluated the performance using evaluation metrics mentioned in section 4.3 on the BraTS 2020 validation dataset that consisted of 125 multimodal brain MR. For our main model, we utilized 5-fold cross-validation and compared the average dice score attained by the model and also inspected the 95 percentile Hausdorff distance attained in each of the tumor subregions. In our experimentation, we used a mean ensemble to predict the online validation dataset of the BraTS 2020 challenge; where the checkpoints stored from each of the fold's runs were used. Due to using the 3D-TC approach in our model, we were able to incorporate a larger patch size of $192 \times 192 \times 128$ as we did not face any GPU memory restrictions. It should be noted without the 3D-TC approach the deeper U-Net model that has been implemented requires 15GB of GPU memory even processing a patch size of $128 \times 128 \times 128$. Along with the evaluation above metrics we also present the parameters number and FLOPs measure of our models. It should be noted that we did not use the DUC decoder-based model in our ensemble inference as it was only trained on a single fold to gauge the memory reduction impact and compare the performance impacts of taking such an approach. We will demonstrate the impact of post-processing from online metrics that were provided by the evaluation website to discuss the gain or loss of post-processing on the predicted output of the ensemble model. For a qualitative analysis of our 3D-TC Deeper U- Net model, the

Table 5.2: Performance comparison of the three implemented models on Fold-2

Model	Mean Dice Score (↑)	Mean HD95 (↓)
3DTC U-Net	0.8260	7.1650
3DTC R2AU-Net	0.8026	9.0912
3DTC DUC U-Net	0.8220	8.1801

Table 5.3: Segmentation performance evaluation of 5-Fold using the Dice score evaluation metric on the training set.

Model	Dice Score (↑)			Mean Dice
	WT	TC	ET	
Fold-1	0.9108	0.8379	0.7838	0.8458
Fold-2	0.8976	0.8235	0.7409	0.8260
Fold-3	0.9122	0.8484	0.7684	0.8446
Fold-4	0.9025	0.8380	0.7867	0.8412
Fold-5	0.9090	0.8688	0.8102	0.8634

evaluation of the validation dataset of BraTS 2020 was compared against previous SOTA(state-of-the-art) methods and conventional segmentation models such as the 3D-U-Net, and V-Net. We presented the performance comparison against the other models in Table 5.4.

Table 5.4: Performance comparison on BraTS 2020 validation set. Per case denote the computational costs of segmenting a 3D patient case

Method	Dice Score (%) ↑			HD95(mm) ↓			Flops
	WT	TC	ET	WT	TC	ET	per case
3D U-Net [42]	84.11	79.06	68.76	13.366	13.607	50.983	1669.53
V-Net [1]	84.63	75.26	61.79	20.407	12.175	47.702	749.29
Deep V-Net [1]	86.11	77.90	68.97	14.499	16.153	43.518	-
Res U-Net [43]	82.46	76.47	71.63	12.337	13.105	37.422	407.37
Liu et al. [44]	88.23	80.12	76.37	6.680	6.490	21.390	-
Vu et al. [45]	90.55	82.67	77.17	4.990	8.630	27.040	-
Ghaffari et al.[46]	90.00	82.00	78.00	-	-	-	-
TransU-Net [47]	89.46	78.37	78.42	5.968	12.840	12.851	1205.76
Swin-UNet [48]	89.34	77.60	**78.95**	7.855	14.594	**11.005**	250.88
TranBTS [49]	89.00	81.36	78.50	6.469	10.468	16.716	333.09
3DTC U-Net	**90.76**	**83.39**	74.77	**4.371**	**6.308**	27.179	303.59

5.4 Online Validation Dataset Results

The distribution of the Dice Co-efficient Score received from the online validation dataset which consisted of 125 cases using the 3DTC-Unet Model is illustrated in Fig. 5.9. We have also illustrated the Median, Standard deviation, 25^{th} and 75^{th} percentile score which we received from the BraTS 20 challenge website. The values attained have been demonstrated in Table. 5.5.

Table 5.5: Quantitative validation set results using the ensembled 3DTC-Unet Model

	Dice			Sensitivity		
	ET	WT	TC	ET	WT	TC
Median	0.8491	0.9243	0.89091	0.862	0.943	0.927
25^{th} quantile	0.74528	0.89256	0.78233	0.74863	0.89686	0.80602
$75^{th} quantile$	0.89348	0.94731	0.93364	0.91387	0.97305	0.96768
StdDev	0.2719	0.06121	0.15678	0.2959	0.07853	0.17855

5.5 Segmented Image Analysis

In this section, we have showcased the predicted segmentation mask made by our model. From Fig. 5.10 and the received result we can gauge that our model was able to make proper inferences on the validation dataset. Our model attained the best mean dice score of (ET: 0.96, WT: 0.97, TC: 0.97). Additionally, for some samples small contrast-enhanced region was falsely predicted as an ET region by our model affecting the ET dice and HD95 metric.

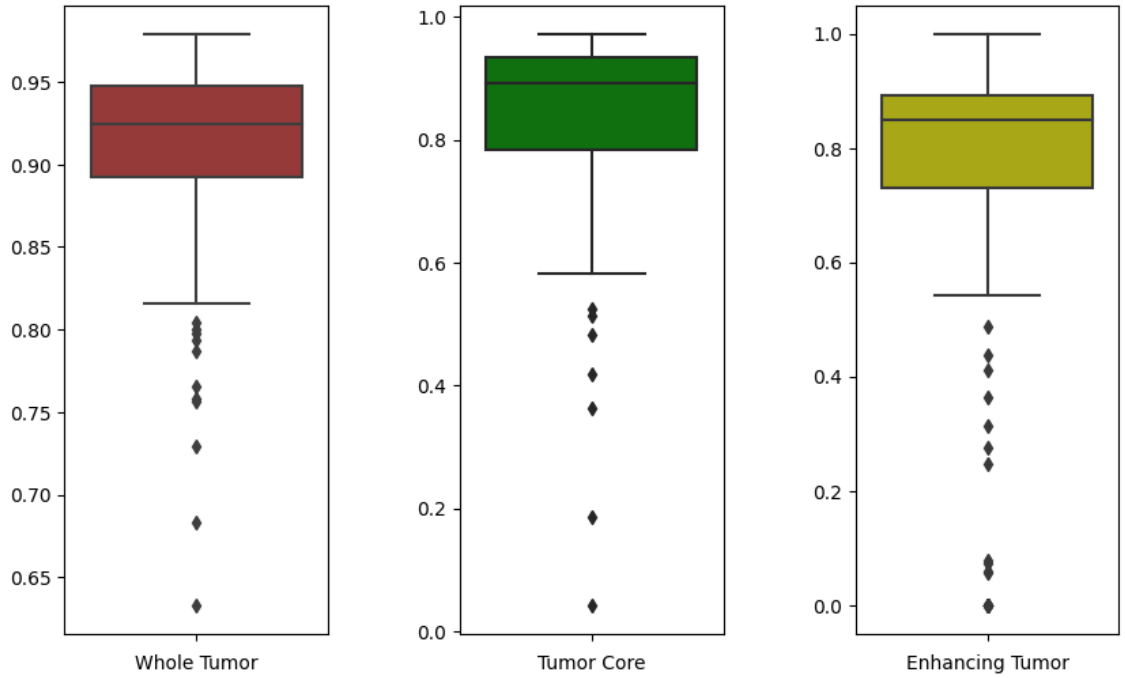

Figure 5.9: Boxplot of the dice score distribution on the three tumor sub-regions on BraTS 2020 validation dataset

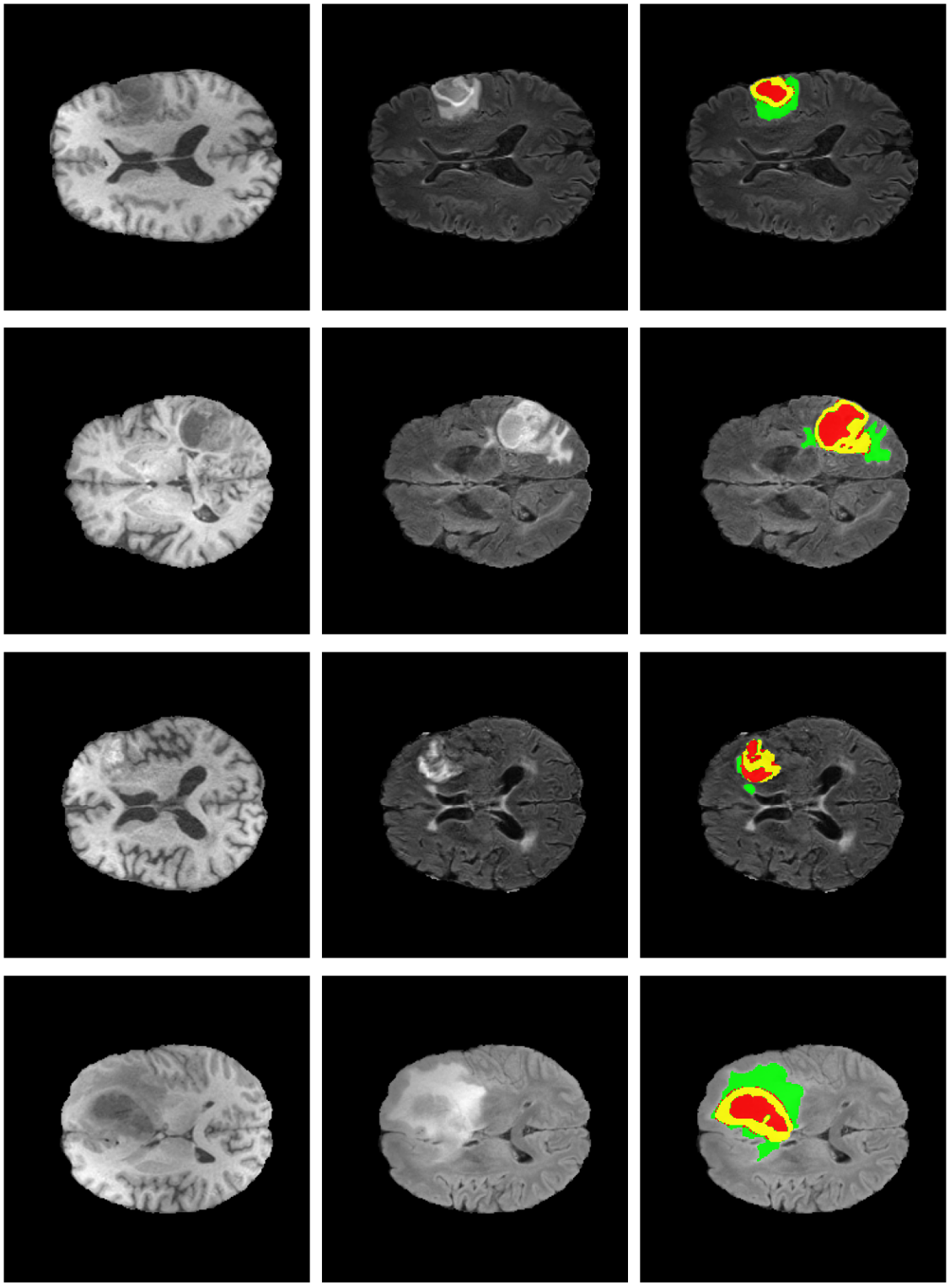

Figure 5.10: Prediction from the Ensembled 3DTC U-Net model on the online validation dataset. The left to rig!htbp column in each row shows the T1, FLAIR, and corresponding annotation respectively.

Chapter 6

Conclusion

To enhance semantic segmentation systems, this research presents a straightforward 3D tiled convolution. The input will first be subjected to a periodic down-shuffling operation before being subjected to the standard 3D convolution in the proposed 3D-TC method. Directly applied to the input data, it benefits from a smaller data size for sub-sequential processing without losing any of the information contained within due to down-shuffling. This study aims to apply the 3D-TC on the input data for reducing the file size. Since there is numerous semantic segmentation done using FCNs we look to address the memory constraint issue in this study and subsequently the performance of the architecture improve is improved. This study will evaluate the network agnostic property 3d tiled convolution & utilize it with other FCNs.

6.1 Future Work

In our research, we employed the U-Net model with the standard VGG-16 backbone non-pretrained backbone which was incorporated with 3D Tiled Convolution and Dense upsampling convolution. In the future, we look to try out the 3DTC approach with other promising models such as DeeplabV3+, BGNet, PCNet, and modern 3D semantic segmentation architectures which currently have complex layer configurations and high GPU memory demand. Along with this, we also want to monitor the effects of different shuffling factors and evaluate the metrics attained for a better summarization of the 3DTC approach. In our experiment we only exploited the 3DTC-based architecture for brain tumor segmentation; in the future, we want to use the 3DTC approach with other biomedical segmentation challenges to reduce computational demand, and time required while maintaining the state-of-the-art performance.

Bibliography

[1] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*, IEEE, 2016, pp. 565–571.

[2] W. Li, G. Wang, L. Fidon, S. Ourselin, M. J. Cardoso, and T. Vercauteren, "On the compactness, efficiency, and representation of 3d convolutional networks: Brain parcellation as a pretext task," in *International conference on information processing in medical imaging*, Springer, 2017, pp. 348–360.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[4] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth, "Medical image segmentation on gpus–a comprehensive review," *Medical image analysis*, vol. 20, no. 1, pp. 1–18, 2015.

[5] G. Zeng and G. Zheng, "3d tiled convolution for effective segmentation of volumetric medical images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 146–154.

[6] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[7] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[8] D. Zikic, Y. Ioannou, M. Brown, and A. Criminisi, "Segmentation of brain tumor tissues with convolutional neural networks," *Proceedings MICCAI-BRATS*, vol. 36, no. 2014, pp. 36–39, 2014.

[9] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, 2019.

[10] A. Jesson and T. Arbel, "Brain tumor segmentation using a 3d fcn with multi-scale loss," in *International MICCAI Brainlesion Workshop*, Springer, 2017, pp. 392–402.

[11] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3d deeply supervised network for automatic liver segmentation from ct volumes," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2016, pp. 149–157.

[12] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, Springer, 2016, pp. 424–432.

[13] T. D. Bui, J. Shin, and T. Moon, "3d densely convolutional networks for volumetric segmentation," *arXiv preprint arXiv:1709.03199*, 2017.

[14] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[15] K. Kamnitsas, C. Ledig, V. F. Newcombe, *et al.*, "Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation," *Medical image analysis*, vol. 36, pp. 61–78, 2017.

[16] M. Kolařík, R. Burget, V. Uher, K. Říha, and M. K. Dutta, "Optimized high resolution 3d dense-u-net network for brain and spine segmentation," *Applied Sciences*, vol. 9, no. 3, p. 404, 2019.

[17] W. Chen, B. Liu, S. Peng, J. Sun, and X. Qiao, "S3d-unet: Separable 3d u-net for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, Springer, 2019, pp. 358–368.

[18] T. Moriya, H. R. Roth, S. Nakamura, *et al.*, "Unsupervised segmentation of 3d medical images based on clustering and deep representation learning," in *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, SPIE, vol. 10578, 2018, pp. 483–489.

[19] J. Chen, Z. Wan, J. Zhang, *et al.*, "Medical image segmentation and reconstruction of prostate tumor based on 3d alexnet," *Computer methods and programs in biomedicine*, vol. 200, p. 105 878, 2021.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[21] F. Isensee, J. Petersen, A. Klein, *et al.*, "Nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.

[22] W. Zhao, D. Jiang, J. P. Queralta, and T. Westerlund, "Mss u-net: 3d segmentation of kidneys and tumors from ct images with a multi-scale supervised u-net," *Informatics in Medicine Unlocked*, vol. 19, p. 100 357, 2020.

[23] Q. Yu, D. Yang, H. Roth, *et al.*, "C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4126–4135.

[24] J. Ngiam, Z. Chen, D. Chia, P. Koh, Q. Le, and A. Ng, "Tiled convolutional neural networks," *Advances in neural information processing systems*, vol. 23, 2010.

[25] G. Zeng and G. Zheng, "Holistic decomposition convolution for effective semantic segmentation of 3d mr images," *arXiv preprint arXiv:1812.09834*, 2018.

[26] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[27] M. Rajchl, M. C. Lee, O. Oktay, *et al.*, "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE transactions on medical imaging*, vol. 36, no. 2, pp. 674–683, 2016.

[28] L. M. Ballestar and V. Vilaplana, "Mri brain tumor segmentation and uncertainty estimation using 3d-unet architectures," in *International MICCAI Brainlesion Workshop*, Springer, 2021, pp. 376–390.

[29] P. Tran, "A fully convolutional neural network for cardiac segmentation in short-axis mri. arxiv 2016," *arXiv preprint arXiv:1604.00494*,

[30] O. Oktay, J. Schlemper, L. L. Folgoc, *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[32] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 472–480.

[33] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *2011 international conference on computer vision*, IEEE, 2011, pp. 2018–2025.

[34] W. Shi, J. Caballero, F. Huszár, *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[35] Q. Zhu, B. Du, B. Turkbey, P. L. Choyke, and P. Yan, "Deeply-supervised cnn for prostate segmentation," in *2017 international joint conference on neural networks (IJCNN)*, IEEE, 2017, pp. 178–184.

[36] B. H. Menze, A. Jakab, S. Bauer, *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015. DOI: 10.1109/TMI.2014.2377694.

[37] S. Bakas, H. Akbari, A. Sotiras, *et al.*, "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features," *Scientific Data*, vol. 4, no. 1, Sep. 2017. DOI: 10.1038/sdata.2017.117. [Online]. Available: https://doi.org/10.1038/sdata.2017.117.

[38] S. Bakas, H. Akbari, A. Sotiras, *et al.*, *Segmentation labels for the pre-operative scans of the tcga-gbm collection*, 2017. DOI: 10.7937/K9/TCIA.2017.KLXWJJ1Q. [Online]. Available: https://wiki.cancerimagingarchive.net/x/KoZyAQ.

[39] S. Bakas, H. Akbari, A. Sotiras, *et al.*, *Segmentation labels for the pre-operative scans of the tcga-lgg collection*, 2017. DOI: 10.7937/K9/TCIA.2017.GJQ7R0EF. [Online]. Available: https://wiki.cancerimagingarchive.net/x/LIZyAQ.

[40] E. Tiu, *Metrics to evaluate your semantic segmentation model*, Oct. 2020. [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2.

[41] S. Du, *Understanding dice loss for crisp boundary detection*, Jul. 2022. [Online]. Available: https://medium.com/ai-salon/understanding-dice-loss-for-crisp-boundary-detection-bb30c2e5f62b.

[42] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2016, pp. 424–432.

[43] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[44] Z. Li, J. Pan, H. Wu, Z. Wen, and J. Qin, "Memory-efficient automatic kidney and tumor segmentation based on non-local context guided 3d u-net," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*, Springer, 2020, pp. 197–206.

[45] M. H. Vu, T. Nyholm, and T. Löfstedt, "Multi-decoder networks with multi-denoising inputs for tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke, and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part I 6*, Springer, 2021, pp. 412–423.

[46] M. Ghaffari, A. Sowmya, and R. Oliver, "Automated brain tumour segmentation using cascaded 3d densely-connected u-net," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke, and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part I 6*, Springer, 2021, pp. 481–491.

[47] J. Chen, Y. Lu, and Q. T. Yu, "Transformers make strong encoders for medical image segmentation. arxiv 2021," *arXiv preprint arXiv:2102.04306*,

[48] H. Cao, Y. Wang, J. Chen, *et al.*, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *European conference on computer vision*, Springer, 2022, pp. 205–218.

[49] W. Wang, C. Chen, M. Ding, H. Yu, S. Zha, and J. Li, "Transbts: Multimodal brain tumor segmentation using transformer," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*, Springer, 2021, pp. 109–119.