

Brain Tumor Detection and Segmentation Using Deep Learning



Abeyu Eshete Tefera

**A Thesis Submitted to the Department of Computer Science
and Engineering,
School of Electrical Engineering and Computing**

**Presented in Partial Fulfillment of Master's Degree in
Computer Science and Engineering**

**Office of Graduate Studies
Adama Science and Technology University**

September 13, 2023

Adama, Ethiopia

Brain Tumor Detection and Segmentation Using Deep Learning

By:

Abeyu Eshete

Advisor:

Getinet Yilma (Ph.D.)

A Thesis Submitted to Department of Computer Science and Engineering,

School of Electrical Engineering and Computing

**Presented in Partial Fulfillment of Master's Degree in
Computer Science and Engineering**

Office of Graduate Studies

Adama Science and Technology University

Sept13, 2023

Adama, Ethiopia

Approval of M.Sc. Thesis

I, the advisors of the thesis entitled “Brain tumor detection and segmentation using Deep Learning” and developed by Abeyu Eshete Tefera, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Getinet Yilma (Ph. D) _____

Major Advisor

Signature

Date

We, the undersigned, members of the Board of Examiners of the thesis by Abeyu Eshete Tefera” have read and evaluated the thesis entitled “Brain tumor detection and segmentation using Deep Learning” and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

Chairperson

Signature

Date

Internal Examiner

Signature

Date

External Examiner

Signature

Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head

Signature

Date

School Dean

Signature

Date

Office of Postgraduate Studies, Dean

Signature

Date

Declaration

I hereby declare that this Master Thesis entitled “Brain tumor detection and segmentation using Deep Learning” is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

<u>Abeyu Eshete</u>	_____	_____
Name of the student	Signature	Date

Recommendation

I, the advisor of this thesis, hereby certify that we have read the revised version of the thesis entitled “Brain Tumor Detection and Segmentation Using Deep Learning” prepared under my/our guidance by Abeyu Eshete submitted in partial fulfillment of the requirements for the degree of Master’s of Science in Computer Science and Engineering. Therefore, I recommend the submission of revised version of the thesis to the department following the applicable procedures.

Getinet Yilma (Ph. D)

Major Advisor

Signature

Date

ACKNOWLEDGEMENT

To begin, I express my utmost gratitude to God. Following that, I extend my sincere appreciation to my Advisor, Getinet Yilma (Ph.D.), whose indispensable guidance, and insightful feedback greatly benefited me due to his extensive expertise. Additionally, I am deeply grateful to my beloved spouse for her unwavering support throughout and my children, through their presence and occasional disruptions, have taught me the virtue of patience. Lastly, I would like to extend a heartfelt thank you to my colleague, Tr. Taye Birhanu, for his consistently motivating me.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ACRONYMS	ix
ABSTRACT.....	x
1. CHAPTER ONE INTRODUCTION.....	1
1.1 Background	1
1.2 Motivation of the Study.....	3
1.3 Problem Statement	3
1.4 Research Questions	4
1.5 Objective	4
1.5.1 General objectives.....	4
1.5.2 Specific Objectives	4
1.6 Significance of the Study	5
1.7 Scope and limitations	5
1.8 Thesis Organization.....	6
2. CHAPTER TWO: REVIEW OF LITERATURE	7
1.9 Brain tumor and Magnetic Resonance Images(MRI)	7
2.1 Deep Neural Network.....	9
2.1.1 Convolution Neural Network.....	11
2.2 Layers in CNN	12
2.4 Related Work.....	18
2.4.1 Summary of Related works	21
2.5 Research Gap.....	24
3. CHAPTER THREE: METHODOLOGY	25
3.1. Overview of the Proposed Methods	25
3.2. Dataset.....	25
3.3. Development and Implementation Tools	26
3.3.1. Hardware tools.....	26
3.3.2. Software tools.....	26

3.4.	Baseline Model.....	27
3.4.1.	DeepUnet.....	27
4.	CHAPTER FOUR: PROPOSED ARCHETICTURE	29
4.1	Residual Unet	30
4.2	Dataset and Its Preprocessing.....	32
4.2.1	Normalization	34
4.2.2	Patching.....	35
4.3	Training the Models	37
4.3.1	Optimizer	37
4.3.2	Loss function.....	37
4.3.3	Regularization	38
4.4	Detection and Segmentation Evaluation Matrices	39
4.5	Summary	41
5.	CHAPTER FIVE: IPLIMENTATION OF THE PROPOSED ARCHETICTUR	42
5.1	Tools of Implementation	42
5.1.1	Hardware Tools	42
5.1.2	Software tools	42
5.2	Training Procedure.....	43
5.3	Proposed Architecture Implementation.....	43
5.3.1	The Encoder Implementation.....	43
5.3.2	The Decoder Implementation	45
6.	CHAPTER SIX: EXPERIMENTAL RESULT AND DISCUSSION	49
6.1	Quantitative Evaluation and Discussion	49
6.1.1	Segmentation Result Analysis	49
6.1.2	Detection Result Analysis.....	51
6.1.3	Detection Rates	53
6.2	Qualitative Experimental Results.....	54
6.3	Training, Validation and Loss Matrices Curves.....	57
6.4	Comparison with State-of-the-art Works	60
6.5	Contribution of the Study	62
6.6	Research question Discussions	63

7. CHAPTERSEVEN:CONCLUSION AND FUTURE WORK.....	65
7.1 Conclusion.....	65
7.2 Future Work	66
Reference	67

LIST OF TABLES

Table 2-1 summary of related works	21
Table 4-1Evaluation matrix	39
Table 5-1Encoder part of the architecture	43
Table 5-2 Decoder Part of the Residual UNet Model.....	45
Table 6-1 State of arts comparison	61

LIST OF FIGURES

Figure 3-1 Generic research process workflow	25
Figure 3-2 The Baseline DeepUnet Architecture.....	28
Figure 4-1 Overview of the Methodology	29
Figure 4-2 The Proposed Residual UNet Architecture	30
Figure 4-3 Shows the architecture of skip connection module.....	31
Figure 4-4 Sample Image from BRATS2020	33
Figure 4-5 Sample Label from BRATS2020	33
Figure 4-6 image Patches.....	36
Figure 4-7 Data Distribution.....	36
Figure 6-1 Residual UNet and Deep UNet performance result	50
Figure 6-2 Residual UNet and DeepUNet detection performance result	51
Figure 6-3. Tumor detection on BRaTs2019	53
Figure 6-4. Tumor Detection Rate on BRaTs2020	54
Figure 6-5 Predicted Results on BRATS2020 dataset.....	55
Figure 6-6 Predicted Results on BRATS2020 dataset.....	55
Figure 6-7 Residual UNet Results on an Image size of 192×192	56
Figure 6-8 Residual UNet Performance using Dice loss	57
Figure 6-9 Residual UNet Learning Curve on BRATS2020 Dataset	58
Figure 6-10 DeepUNet Learning Curve on BRATS2020 Dataset.....	58
Figure 6-11 Residual Unet Learning Curve on BRATS2019 Dataset.....	59
Figure 6-12 Deep Unet Learning Curve on BRATS2019 Dataset	60

LIST OF ACRONYMS

CNS	Central nervous system
DL	Deep learning
CNN	Convolution neural network
MRI	Magnetic resonance imaging
DNN	Deep Neural Network
FC	Fractional Calculus
SO	Snake Optimizer
RNNs	Recurrent neural networks
ReLU	Rectified linear unit
DSS	Dice similarity score
NLP	Neural Language processing

ABSTRACT

Brain tumor detection and segmentation refer to the processes of identifying the presence of brain tumors in medical imaging data and accurately delineating the tumor boundaries within the images. These processes involve classifying the pixels within the image as tumor or non-tumor regions. Accurate segmentation is essential for quantifying tumor size, analyzing tumor characteristics, monitoring tumor progression, and assisting in treatment planning. Brain tumor detection and segmentation are widely used techniques in brain magnetic resonance imaging (MRI) analysis for quantifying and visually representing the anatomical structures of the brain, detecting pathological regions, planning medical procedures, and guiding treatments. Due to the wide variety in the size, shape, location, and appearance of brain tumors, brain tumor detection and segmentation are challenging tasks that needs extensive experimental investigation. Deep Neural Network algorithms like UNet outperform many semantic detection and segmentation tasks. However, Unet's shallow networks limit the model's ability to extract abstract and global contextual features. Therefore, this thesis presented two improved Unet architectures, that are DeepUnet as a baseline and Residual Unet as a proposed model, which capture multiscale features to enable the models to extract both local and global contextual features. The proposed method is hypothesized to capture important feature of non-uniform shapes, sizes and location from MRI images. Residual Unet, enhances the Unet architecture by incorporating a residual and skip connection module to handle both local and global features, as well as increasing the depth of the encoder network to extract more abstract features. These improvements resulted in higher detection and segmentation accuracy, providing clinicians with more accurate diagnostic and treatment planning information for brain tumor patients. Experimental results showed that the proposed model Residual Unet has achieved an accuracy of 99.33% on Brats 2020, and 98.75% on Brats 2019. On the other hand Deep Unet which is the baseline for the proposed model has achieved an accuracy of 99.11% on Brats 2020 and 98.58% on Brats 2019. So Residual model has outperformed better than baseline model which is Deep Unet in terms of performance. These findings indicate that the proposed models can greatly improve and facilitate brain tumor detection and segmentation.

Keywords: Magnetic Resonance Imaging, Deep Neural Network, Deep Learning, Convolutional Neural Network, Deep Unet and Residual Unet

1. CHAPTER ONE: INTRODUCTION

1.1 Background

The brain is an essential organ that serves as the command center for controlling bodily functions and cognitive processes. It regulates both voluntary and involuntary actions, playing a pivotal role in decision-making. However, the occurrence of an abnormal tissue growth, referred to as a brain tumor, can disrupt the normal functioning of this complex organ. Tumors in the brain arise from the uncontrolled and excessive proliferation of unwanted fibrous tissue, leading to various disturbances in brain function (P Gokila Brindha,M Kavindraaj ,P Manivasakam,and P Prasanth, 2020).

A brain tumor refers to an irregular proliferation of brain cells that occurs in an unmanageable manner. These tumors can be either cancerous or noncancerous. The pressure inside the skull can expedite the progression of a brain tumor. In the most severe instances, it can lead to brain impairment, posing a significant risk to one's life. According to a projection, approximately 18,020 adults were expected to succumb to primary cancerous brain and central nervous system (CNS) tumors in the year 2020 (Mohammad Attique Kahan,Imran Ashraf,Majed Alhaisoni,ertas Damaševičius,Rafal Scherer,Amjad Rehman,Syed Ahmad Chan Bukhari, 2020). MRI data reveals distinct features and categories of brain tumors, leading to their identification and segmentation by experts. Consequently, MRI scans are commonly employed to aid in the detection and delineation of brain tumors. By utilizing MRI images, doctors can assess tumors and devise treatment strategies accordingly. The treatment approach relies on multiple factors, such as the tumor's shape, size, type, grade, and location, all of which can vary significantly depending on the patient's condition. Therefore, precise recognition and classification of brain tumors plays a crucial role in ensuring appropriate treatment (Mohammad Attique Kahan,Imran Ashraf,Majed Alhaisoni,ertas Damaševičius,Rafal Scherer,Amjad Rehman,Syed Ahmad Chan Bukhari, 2020).

Deep learning, a branch of machine learning, concentrates on training artificial neural networks to acquire knowledge and make informed choices or predictions. Particularly in the realm of medical image analysis, deep learning has made notable advancements, including the detection and segmentation of brain tumors. This progress is primarily attributed to the greater accessibility of

medical images and the rising demand for precise and efficient techniques for disease identification and segmentation(Hedao, 2021).Before the emergence of deep neural networks (DNNs), conventional approaches were regarded as the most effective methods for brain tumor detection and segmentation. These methods heavily relied on manually designed features to classify individual pixels. The choice and extraction of these handcrafted features were crucial in the process (Amjad Rehman Khan, Siraj Khan, Majid Harouni, Rashid Abbasi, Sajid Iqbal, Zahid Mehmood, 2021)is inspired by domain expertise and observations of tumoral image region in various MRI modalities. However, nothing guarantees that these extracted features will be a good description for the image to be detected and or segmented. The relevancy of features and the chosen machine learning techniques have a big impact on the segmentation quality, generalizability and tumor region visualization. The more relevant they are, the higher the segmentation quality and detection rate will be (Hedao, 2021).

Unlike traditional machine learning, deep neural network models perform automatic feature extraction by determining the appropriate set of features for a better detection outcome on their own(Schmidhuber, 2015) and can also be trained end-to-end which makes them popular in detection and segmentation tasks. Deep structured learning, often known as deep learning or hierarchical learning, has been a growing topic of machine learning research with in the past two decades. The research techniques have improved significantly during the last few years. Deep neural network research has already had an impact on a wide range of signal, image and information processing activity (Schmidhuber, 2015).

The proposed method in this thesis can employ deep learning technique to improve detection and segmentation of brain tumor. The technique is expected to minimize the risks of misdiagnosis and mistreatment, which can have a significant impact on patient treatment outcomes. By improving the detection of the tumorous region of the brain and distinguishing it from healthy tissue, these methods can aid medical doctors in diagnosing the disease more easily. The manual segmentation of brain tumors is an essential procedure that necessitates a team of clinical specialists to accurately identify the location and type of the tumor. Additionally, it is a labor-intensive process that relies on the physicians' subjective experience, skills, and qualifications. Physicians must carefully label each slice of the tumor; as larger nodules often extend over multiple slices. Modeling the treatment of tumors poses challenges and consumes significant time due to the irregular shapes of tumors and their varied locations.

Therefore, the significance of an automated brain tumor detection and segmentation method lies in its ability to assist doctors in interpreting medical images and making decisions about stages of tumor growth. Deep neural network techniques are particularly effective in detecting timorous and segmenting regions, and the proposed methods have great significance in aiding computer-aided brain tumor diagnosis, which is a critical first step in many clinical applications before further treatment.

1.2 Motivation of the Study

Developing more accurate and efficient methods for detecting and segmenting brain tumors could have several motivations. Primarily, it has the potential to enhance the precision and effectiveness of brain tumor diagnosis, ultimately resulting in improved outcomes for patients. Secondly, it could help doctors develop personalized treatment plans for patients. Thirdly, it could lead to advancements in medical imaging technology more broadly, with potential applications in other areas of medicine. Lastly, studying brain tumor detection and segmentation could help us gain a better understanding of these tumors, which could ultimately lead to new treatments and cures.

1.3 Problem Statement

The detection and segmentation of brain tumors are of utmost importance in medical imaging analysis, as they provide vital information for the diagnosis and treatment of these conditions. However, this task is exceptionally challenging due to the complex characteristics of brain tumors, which exhibit diverse shapes and sizes. Moreover, manually identifying and delineating tumor regions in the brain is a laborious and time-consuming process that demands a high level of expertise and proficiency.

To address these obstacles, scholars have concentrated on creating automated and effective approaches for brain tumor detection and segmentation, leveraging innovative machine learning and deep learning methodologies. These techniques hold the promise of delivering precise and prompt diagnoses and treatment strategies for individuals with brain tumors. In recent times, the integration of deep learning methods, like convolutional neural networks, has shown encouraging outcomes in this domain.

However, even with the successes achieved using deep learning methods, the challenge persists in the task of detecting and segmenting brain tumors due to uncertainties in tumor locations, variations in tumor shapes and sizes, and imbalanced data. Furthermore, neural networks with limited receptive fields struggle to capture global contexts, as they can only gather local information within a limited spatial scope. As a result, there is a pressing requirement to explore novel methodologies and approaches that can enhance the precision and efficiency of brain tumor detection and segmentation.

1.4 Research Questions

The upcoming research will address the following questions:

1. How does Residual UNet model improve the efficiency of brain tumor detection and segmentation compared to traditional methods?
2. What benefits does employ different datasets offer when training and evaluating deep neural network models in the context of brain tumor detection and segmentation?
3. What performance improvements can be achieved compared to other state-of-art works?

1.5 Objective

1.5.1 General objectives

The general objective of this thesis is to propose a deep learning-based brain tumor detection and segmentation method for improved diagnosis and treatment recommendation.

1.5.2 Specific Objectives

1. To propose effective brain tumor detection and segmentation method using different network architectures, such as Deep UNet and Residual UNet network architectures for improved brain tumor detection and segmentation.
2. To train on different bench mark datasets for evaluating the proposed method for brain tumor detection and segmentation.
3. To evaluate the performance of the proposed method with different detection and segmentation metrics on existing benchmark datasets.
4. Compare the proposed method with other state-of-art related works.

1.6 Significance of the Study

The proposed methods make notable contributions to the advancement of brain tumor detection and segmentation, ultimately enhancing the diagnosis of patients with brain tumors.

1. Improved tumor detection and segmentation: Detecting even the smallest tumor regions and accurately segmenting the tumor boundaries, leads to better diagnosis and treatment planning.
2. Time Efficiency: Reduce the time required for diagnosis and treatment.
3. Generalizable and reusable: Deep learning-based algorithms subjective variability, ensuring consistent and reliable tumor detection and segmentation across different medical professionals and institutions.

1.7 Scope and limitations

This study is centered around the detection and segmentation of brain tumors using magnetic resonance imaging (MRI) data, employing the DeepUnet as a baseline and Residual Unet deep learning as a proposed model. The dataset will consist of diverse cases, encompassing various brain tumor types, sizes, and locations, to evaluate the accuracy and comparative performance of both DeepUnet and Residual Unet for whole tumor detection and segmentation. We will limit the annotation level to whole tumor detection and segmentation. Pre-processing techniques, including normalization and noise reduction, will ensure consistent data preparation. The study will utilize available computational resources for training and validation of the DeepUnet and Residual Unet models. Evaluation of the detection and segmentation results will be performed using metrics like Dice coefficient and sensitivity. Ethical considerations will be addressed concerning data privacy and informed consent for using patient data.

This study on brain tumor detection and segmentation using the DeepUnet and Residual Unet deep learning architectures faces several limitations. First, the availability of high-quality annotated brain tumor MRI data may be limited, potentially introducing data imbalances and biases. Moreover, generalizing the models to rare or uncommon tumor types and locations not well-represented in the training data could be challenging. Inter-observer variability in manual annotations might impact the reliability of ground truth labels. The risk of over fitting due to limited datasets necessitates careful regularization and cross-validation. Additionally, the complexity and interpretability of deep learning models raise concerns regarding their black-box

nature. The computational demands of training these models and the time-consuming hyper parameter tuning further constrain the study. Limited clinical validation and the presence of heterogeneous and irregular tumor shapes add complexity to accurate segmentation. Compatibility issues with existing medical systems and ethical considerations concerning data privacy and informed consent must be addressed for real-world deployment. Addressing these limitations throughout the study ensures the robustness and practical applicability of the brain tumor segmentation models in clinical settings.

1.8 Thesis Organization

In Chapter 1, a thorough introduction was presented, highlighting the importance of deep neural networks in brain tumor detection and segmentation. The chapter extensively examines the existing literature on this topic, providing a detailed problem statement, as well as outlining the research objectives and contributions.

In Chapter 2, the fundamental theoretical concepts crucial for comprehending the applied methodology in the study are introduced. The chapter presents a thorough examination of brain tumor detection and segmentation, encompassing the fundamental components and principles of deep neural networks (DNN).

In Chapter 3, a concise description of the proposed methodologies and the experimental arrangement is presented, encompassing information on the utilized dataset, preprocessing procedures, hyper parameters employed for model training, and the evaluation metrics applied in this research.

Chapter 4 Demonstrates the qualitative and quantitative experimental outcomes, discussions, and evaluations of the model's performance is conducted using diverse evaluation metrics.

Chapter 5 marks the conclusion of our study and outlines possible avenues for future research.

2. CHAPTER TWO: REVIEW OF LITERATURE

1.9 Brain tumor and Magnetic Resonance Images(MRI)

A brain tumor refers to an irregular proliferation of brain cells that occurs in an unmanageable manner. The structure of the brain is highly intricate, consisting of various regions that perform distinct functions within the nervous system. Brain tumors have the potential to occur in different parts of the brain and skull, including the protective lining, skull base, brainstem, sinuses, nasal cavity, and other locations. They can originate from various tissue types, leading to a diverse range of more than 120 distinct tumor types that can form within the brain(The Johns Hopkins University, The Johns Hopkins Hospital, and Johns Hopkins Health System, 2022).

The manual identification of brain tumors through MRI images requires significant computational resources, especially in cases where prompt treatment is critical for patient well-being and accuracy heavily relies on specialized knowledge. Detecting tumors continues to be a formidable task due to notable variations in their location and structure, which are typically characterized by irregular shapes and ambiguous boundaries. (Masood M, Nazir T, Nawaz M, Mehmood A, Rashid J, Kwon HY, Mahmood T, Hussain A., 2021).

Brain tumor segmentation is a task in medical image analysis that focuses on distinguishing brain tumors from healthy brain tissue in magnetic resonance imaging (MRI) scans. The objective is to generate a segmentation map, either binary or multi-class, that precisely outlines the tumor's location and size. By achieving accurate brain tumor segmentation, medical professionals can better understand the tumor's characteristics and plan appropriate treatment strategies. Brain tumors encompass some of the most dangerous tumor types worldwide. Among them, glioma stands out as the most prevalent primary brain tumor, originating from the abnormal growth of glial cells in the brain and spinal cord (Ranjbarzadeh, R., Bagherian Kasgari, A., Jafarzadeh Ghouschi, S. et al, 2021). Magnetic Resonance Imaging (MRI) is a commonly employed non-invasive technique that generates a vast array of tissue contrasts within each imaging modality. It has gained significant popularity among medical professionals for diagnosing brain tumors. The diverse and extensive information provided by MRI scans allows specialists to accurately assess and identify brain tumor conditions (Ranjbarzadeh, R., Bagherian Kasgari, A., Jafarzadeh Ghouschi, S. et al, 2021).

Segmentation is the process of dividing a given image into multiple homogeneous segments. By representing an image in simpler and more manageable forms, segmentation enhances the effectiveness of pattern recognition. Image segmentation (Salwa Khalid Abdulateef and Mohanad Dawood Salman, 2021) is the process of splitting an image into homogeneous regions based on particular criteria. Each region has the same characteristics in terms of a certain attribute. In particular, brain tumor segmentation (Al-Qazzaz, S., 2020) is the process of identifying and separating the region of brain tissue affected by a tumor from the surrounding healthy tissue.

Lately, there has been a noticeable shift in the field of brain tumor segmentation, favoring the adoption of deep neural network techniques. These methods have demonstrated remarkable promise in surpassing the constraints of conventional approaches and attaining remarkable levels of accuracy and efficiency in brain tumor segmentation. Furthermore, the continuous progress in deep neural network research has led to the introduction of more sophisticated techniques like multi-modal and multi-scale networks, which contribute to further improvements in the quality of segmentation results.

Moreover, there is a growing inclination to employ larger and more varied datasets to train and assess deep neural network models. The adoption of larger datasets offers a broader spectrum of brain tumor variability, enabling the model to improve learning and generalization. Additionally, the utilization of multi-modal datasets empowers deep neural network models to leverage complementary information from diverse imaging modalities, leading to enhanced segmentation accuracy.

The Progress of magnetic resonance imaging (MRI) (Grover, V. P., Tognarelli, J. M., Crossey, M. M., Cox, I. J., Taylor-Robinson, S. D., & McPhail, M. J., 2015) to be utilized in medical field has significantly advanced diagnosis, especially with the avoidance of exposure to hazardous ionizing radiation. MRI is especially valuable for visualizing the brain as it provides high-resolution images with good contrast between different tissues, making it easier to identify and segment brain tumors. When radio waves are applied, the protons are disrupted and the energy they emit is captured by the MRI machine and processed to create images.

To obtain a detailed spatial representation of the organ, the MRI machine creates contrast between different layers of human tissue. Subsequently, the resulting MRI images can be observed from

various planes, including axial, coronal, and sagittal, offering a comprehensive view of the brain. Presently, MRI imaging has become the standard procedure for assessing brain tumors, enabling the evaluation of the pathological condition of brain tissue. Unlike X-Ray and CT imaging methods, MRI employs a noninvasive approach that provides high-quality images without causing damage or producing skull artifacts. It offers a complete understanding of the anatomical structure with excellent resolution for soft tissue (Xue, Z., Jiang, M., Liu, H., Zeng, S., & Hao, J., 2020).

Various angles of MRI can produce different sequences of the same tissue, resulting in multi-modal MRI images with varying contrasts. Four frequently employed MRI modalities in brain imaging studies are T1-weighted (T1), T2-weighted (T2), T1-weighted image contrast (T1ce), and fluid attenuated inversion recovery (FLAIR) (Lingling Fang, Xin Wang, 2022). T1-weighted imaging is well-suited for distinguishing different tissues and is particularly useful in identifying brain tumors. It also aids in visualizing the brain's anatomy and surrounding tissues. T1-weighted imaging with contrast enhancement involves injecting a contrast agent into the bloodstream to highlight abnormal brain tissue areas. This method is valuable in identifying malignant tumors, which exhibit higher blood flow and appear brighter on the contrast-enhanced images. T2-weighted imaging emphasizes fluid-filled structures, such as edema and cysts, which are important for diagnosing brain tumors and differentiating healthy and abnormal tissues. Fluid Attenuated Inversion Recovery (FLAIR) is especially effective in identifying white matter lesions, commonly associated with conditions like multiple sclerosis. FLAIR images are also utilized to detect brain changes related to aging and other medical conditions.

2.1 Deep Neural Network

Deep Learning Algorithms (DLAs) belong to the realm of artificial intelligence and machine learning, seeking to emulate human knowledge acquisition. They represent an evolution of traditional neural networks into sophisticated Deep Neural Network (DNN) approaches. These networks possess data-driven connections and autonomously generate methodologies without external intervention, leading to impressive precision and performance in diverse fields. DLAs comprise multiple nerve-based algorithms that independently identify features and attributes within input data before leveraging that knowledge to devise interventions (Ayesha Younis, Li Qiang Charles Okanda Nyatega, Mohammed Jajere Adamu and Halima Bello Kawuwa, 2022).

Deep neural network (Hedao, 2021) is a type of machine learning algorithm that has gained significant attention for brain tumor detection and segmentation because of its capacity to autonomously extract features from extensive datasets.

The deep neural network approach is recognized as the Gold Standard by the machine learning community and is increasingly embraced in the domain of machine learning. This is because DNNs have achieved remarkable results in challenging tasks, that are on level with or even exceeding human performance. For example, deep neural networks have been shown to outperform humans in tasks such as image segmentation, recognition and natural language processing.

One of the key advantages of deep neural networks is their ability to learn from massive amounts of data. This is particularly important in fields such as medical image processing, where large datasets are often required to train accurate models. In recent years, the application of deep neural networks has demonstrated great success in various fields such as bioinformatics, robotics, marketing, security, information retrieval, self-driving vehicles, mobile communications, networking, drug discovery, and more.

Within the realm of medical image processing, DNNs have found application in tasks like brain tumor detection and segmentation, displaying exceptional performance comparable to the latest advancements. Anticipated to gain further traction, the adoption of deep neural networks is foreseen to expand in the future as researchers uncover an increasing number of applications for this potent machine learning methodology.

Deep neural networks, categorized as a distinct variant of artificial neural networks (ANNs), are purposefully constructed to handle both structured and unstructured data. Consisting of an input layer, an output layer, and numerous hidden layers, each layer encompasses nodes responsible for conducting computations on the input data. By utilizing intricate mathematical functions, the neurons within the hidden layers transform the input data, empowering the network to discern intricate patterns and relationships that traditional approaches frequently fail to uncover.

Deep Neural Networks (DNNs) have the ability to grasp non-linear connections within data, enabling them to identify intricate relationships between various variables that might not be easily noticeable to humans. This capability is achieved through iterative learning, wherein the network

is exposed to extensive data during training, progressively enhancing its proficiency in recognizing patterns and making precise predictions.

Regarding brain tumor segmentation, two widely employed deep neural network (DNN) structures are CNNs and RNNs. CNNs are specially crafted for image data processing, making them exceptionally proficient at recognizing patterns in medical images like brain scans. On the other hand, RNNs excel in handling sequential data, like time-series data from medical sensors. Leveraging these DNN architectures, researchers and medical practitioners can greatly enhance their capacity to precisely detect and diagnose brain tumors, resulting in improved health outcomes for patients. The following section will present a concise overview of the Convolutional Neural Network (CNN).

2.1.1 Convolution Neural Network

CNN (Convolutional Neural Network) is a type of neural network designed to process data with a grid-like structure. The convolution operation, performed in the convolutional layer, applies a linear algebra operation by multiplying a filter matrix with the input image to be processed. The convolutional layer is a fundamental and essential layer in CNN. Another frequently utilized layer is the pooling layer, which extracts the maximum or average values from designated regions of the image. CNN can learn intricate features by constructing feature maps. The convolutional layer utilizes kernels that slide over the input sample, generating multiple feature maps. In the input sample, features are identified and depicted as small boxes on the feature map. These maps are then forwarded to the pooling layer, responsible for retaining meaningful features while discarding others. The features preserved by the pooling layer are converted into a one-dimensional feature vector within the fully connected layer. This vector is then utilized to compute the output probability (Dr. J. Jegathesh Amalraj, S. Banumathi, J. Jereena John, 2019).

The user's input is required for three primary purposes: initialization, intervention or feedback response, and evaluation. Initialization involves defining a region of interest (ROI) to restrict the algorithm's processing to the estimated tumor area. Additionally, parameters of pre-processing techniques can be adjusted to suit the input images. Apart from initialization, automated algorithms can be guided towards the desired outcome throughout the process by receiving feedback. This

iterative process allows for adjustments and improvements based on the feedback received (Arkapravo Chattopadhyay, 2022).

A convolutional neural network (CNN) is a highly effective approach for image processing, specifically tailored for tasks such as image recognition and processing. It is a specialized form of artificial neural network that focuses on processing pixel data. The distinguishing characteristic of CNNs is the utilization of a mathematical operation called convolution, which is a specialized type of linear operation. In convolutional networks, convolution is employed instead of general matrix multiplication in at least one of the network layers, making them distinct from traditional neural networks (Hamza Rafiq Almadhoun and Samy S. Abu Naser, 2022).

2.2 Layers in CNN

A CNN comprises three primary layer types: The key components of a convolutional neural network (CNN) include convolutional layers, pooling layers, and fully connected (dense) layers. Moreover, activation layers are incorporated after each convolutional and fully connected layer.

2.2.1 Convolutional layer

There can be multiple convolutional layers in a CNN. The convolutional layer which is the first layer takes the images as input and initiates the processing. Its objective is Extract a set of features from the image while maintaining relationships between the nearby pixels.

The filters used in the convolution layer are designed to detect specific pattern sat different positions and scales in the image. During the convolution process, the filters are convolved with the input image by sliding them across the image and computing the dot product between the filter weights and the corresponding pixel values of the input image (Sakshi Indolia,Anil Kumar Goswami,S. P. Mishra,Pooja Asopa, 2018). During this operation, a single value is generated and assigned to the corresponding position in the output feature map. By employing multiple filters, the convolutional layer can extract numerous features from the input image, resulting in multiple feature maps. The number of filters employed in the convolutional layer determines the quantity of output feature maps produced. Each feature map emphasizes a distinct characteristic of the input image, such as edges, corners, or textures.

The mathematical expression for the convolution operation can be represented as follows.:

$$C[i, j] = (f * g)[i, j] \dots \dots \dots 2.1$$

Where $C[i, j]$ is the output value at location (obj) in the feature map, f is a convolution kernel, g is the input image, and $*$ represents the convolution operation

Instead of being represented as a simple list of numbers, the image is expressed as a collection of small matrices containing pixel values. These matrices, known as input tensors, are multiplied with convolutional kernels (weight matrices) of the same size. By employing various kernels, it is possible to extract different features from the same image. The results are placed at their corresponding spatial positions in the output, forming an output tensor or feature map.

To achieve this, a scanning window, with a fixed number of pixels called stride, is moved across the input image, and the convolution is performed for each window using the same kernel. When the strides are increased, the number of overlapping windows decreases, leading to an output with reduced spatial dimensions.

2.2.2 Pooling Layer

Pooling layers are the second type of layer used in a CNN. There can be multiple pooling layers in a CNN. Each convolution layer is followed by a pooling layer.

The following are Objectives pooling layer:

1. Extract the most significant features by either selecting the maximum value or calculating the average value.
2. Reduce the dimensionality (number of pixels) of the output returned from previous convolution layers.
3. Diminish the quantity of parameters within the network.
4. Remove any noise present in the features extracted by previous convolutional layers.
5. Increase the accuracy of CNNs.

There are three elements in the pooling layer: Feature map, Filter and Pooled feature map. The *pooling operation* occurs in each pooling layer.

The pooling operation happens between a section of the feature map and the filter. It outputs the pooled feature map.

There are two types of pooling operations.

- **Max pooling:** Obtain the maximum value within the region where the filter is applied.
- **Average pooling:** Calculate the average of the values within the region where the filter is applied.

2.2.3 Activation function

Activation function is a vital component of a neural network that helps the model learn complex patterns by adding non-linearity to the output of the layer. By supplying a gradient along with the error, it facilitates the back-propagation process, allowing for the updating of bias and weight parameters. The activation function takes the normalized output of the previous layer and converts it into a special form that can be used as input for the next layer. This function is critical for introducing non-linearity into the neural network, allowing it to capture complex relationships between the input and output.

2.2.4 Batch Normalization

Batch normalization is a technique implemented in deep neural networks to improve the training process and enhance the overall performance of the network. The difficulty in training deep learning models with multiple layers arises from their susceptibility to initial random weights and learning algorithm setups. Batch normalization's fundamental concept is to standardize the inputs for each layer in the network during training. This is achieved by subtracting the mean and dividing by the standard deviation of the inputs within each mini-batch. This normalization process aids in stabilizing the input distribution for each layer, effectively mitigating the internal covariate shift issue that can arise during training.

Given a mini-batch of size m , let $x_{\{1\}}, \dots, x_{\{m\}}$ be the input vectors to the current layer, and γ and β be the two learnable parameters, shifting and translation parameters, learned during the training. The normalized version of $x_{\{i\}}$ is calculated as follow.

$$\bar{x}_i = \frac{x_i - \left(\frac{1}{m} \sum_{i=1}^m x_i\right)}{\sqrt{\left(\frac{1}{m} \sum_{i=1}^m (x_i - \left(\frac{1}{m} \sum_{i=1}^m x_i\right))^2 - \varepsilon\right)}} \dots \dots \dots 2.2$$

Then, the batch normalization(Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q, 2018) operation, y_{et} , is defined as:

$$y_i = \gamma \bar{x}_i + \beta \dots \dots \dots 2.3$$

Batch normalization can be a useful method to improve the performance of deep neural networks in brain tumor segmentation, particularly in challenging scenarios where there is high variability and noise in the data.

2.2.5 Dropout Layer

To tackle overfitting and improve generalization, dropout is utilized as a regularization technique in deep neural networks. It functions by randomly excluding a group of neurons from a layer during the training process. Usually positioned after a fully connected or convolutional layer in the neural network structure, the dropout layer receives activations from the preceding layer as input and stochastically disregards a fraction of these activations based on a predefined probability.

2.2.6 Fully connected Layer

A fully connected layer, also referred to as a dense layer or fully connected feed-forward layer, is a neural network layer in which each neuron is connected to every neuron in the preceding layer. Its primary purpose is to leverage the high-level features extracted from the preceding layers to make predictions based on the input data.

These layers play a concluding role in a CNN architecture. They receive the flattened input from the preceding layer and can include multiple fully connected layers. The final layer is responsible for carrying out classification or other relevant tasks. Each fully connected layer employs an activation function. The main goal of the fully connected layer is to classify the identified features in the image into specific class labels.

The fully connected layer produces an output vector of the same length as the number of output neurons, typically representing the classes in a classification task. Each element in the vector

signifies the probability of the input belonging to a specific class. Since each neuron has its own set of weights and biases, the fully connected layer entails a significant number of parameters. As a result, training this layer can be computationally demanding and time-consuming, particularly when dealing with deep neural networks that have numerous layers.

2.3 Unet and Resnet

UNet: UNet is a convolutional neural network architecture (Olaf Ronneberger, Philipp Fischer, Thomas Brox, 2015). It was designed for biomedical image segmentation but has since found applications in various other domains as well. The name "UNet" comes from the U-shaped architecture it exhibits. The network consists of an encoder and a decoder part.

Encoder: The encoder component resembles a standard convolutional neural network, employing convolutional layers to extract high-level features from the input image. As the encoder progresses, it progressively reduces the spatial dimensions of the data while simultaneously increasing the number of feature channels (Olaf Ronneberger, Philipp Fischer, Thomas Brox, 2015).

MVSS-NET excels in leveraging both the explicitly extracted boundary artifacts and holistic information from multiple views in a seamless manner. Its multi-view feature learning module is purposefully designed to extract features that are independent of specific semantic contexts, allowing them to be applied and adjusted to diverse scenarios (Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao and Xirong Li, Member, 2022)

Decoder: The decoder segment receives the features generated by the encoder and performs upsampling to restore them to the original dimensions of the input image. This upsampling is accomplished by utilizing transposed convolution (also known as deconvolution or upsampling) layers. The primary objective of the decoder is to produce a segmentation map that matches the size of the input image. In this segmentation map, each pixel corresponds to a specific class label (Olaf Ronneberger, Philipp Fischer, Thomas Brox, 2015).

The cascaded Unet model employs a deep network framework where the initial segmentation focuses on the entire tumor, followed by segmentation of internal substructures. Recognizing that deeper layer depths resulting from cascade structures can compromise accurate localization information, we address this by incorporating numerous skip connections. These connections link

features at the same resolution, facilitating the transmission of detailed information from shallow layers to deeper layers (Hongying Liu, Xiongjie Shen, Fanhua Shang, Fei Wang, 2019).

A new deep neural network, named as ‘Enhanced Encoder AtrousUnet’ (EEA Unet) (Sathananthavathi V., Indumathi G., 2021) for retinal vessels segmentation. The proposed architecture is derived from the Unet (Ronneberger et al., 2015). Unet is a widely used U-shaped architecture designed for pixel-wise segmentation of medical images, particularly in the field of medical image analysis. Unet features an encoder-decoder structure, where the number of kernels increases as the depth of the encoder and decoder stages progresses. However, despite its popularity, extracting thin blood vessels remains a challenging problem.

To address this issue, numerous research studies have sought to enhance retinal vessel extraction by introducing modifications to the Unet architecture. The MECU-Net model significantly reduces the number of down-sampling channels. To compensate for the potential loss of accuracy resulting from this reduction, the model employs a cascade strategy, multi-scale information fusion mechanism, and a combination of edge loss and weighted dice loss. These strategies aim to mitigate the oscillation problem during model training and effectively enhance the segmentation performance, particularly for small tumors (Xinchao Cheng, Zongkang Jiang, 2019).

The architecture is designed to capture both global context information from the encoder and detailed localization information from the decoder. This combination makes UNet effective for various segmentation tasks.

Residual UNet: ResidualUNet is an extension of the UNet architecture that incorporates residual connections from the ResNet architecture. Residual connections, introduced in the ResNet model, help alleviate the issue of the gradient vanishing in extremely deep networks, allowing for the training of deeper models. The Resnet-UNet-Fractional Snake Optimization Algorithm (Res-UNet-FSOA) is proposed as a method for segmenting cranial nerves. The process begins by taking MRI images as input and applying preprocessing steps, which include median filtering. Within the preprocessing module, an improved multiscale vessel Ness method is used to enhance the images, specifically detecting local tubular structures. The cranial nerve segmentation is then performed using Res-UNet, which combines elements from Resnet and UNet architectures. Training the network involves employing the Fractional Snake Optimization Algorithm (FSOA), which

integrates Fractional Calculus (FC) and Snake Optimizer (SO) techniques. Start point and end point extraction is achieved through deep seeded region growing (DSRG), and the medial axis extraction is carried out using the tensor voting and non-maximum suppression (TV-NMS) method. Residual UNet, a specialized variant of the UNet architecture, is utilized in Res-UNet, incorporating residual blocks inspired by ResNet. These blocks, consisting of multiple convolutional layers, effectively address the challenge of vanishing gradients in deep neural networks (Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015).

Deep residual networks demonstrate ease of optimization, in contrast to plain networks that rely on straightforward layer stacking. As the depth of the networks increases, plain networks tend to exhibit higher training errors. To tackle the degradation problem, a deep residual learning framework is introduced. Instead of relying on the assumption that a few stacked layers can directly learn the desired underlying mapping, the approach explicitly allows these layers to learn a residual mapping. (Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015).

By integrating residual connections into the UNet, the ResUNet can handle deeper architectures more effectively. The residual blocks in ResUNet help in the smooth flow of gradients during training, making it easier to optimize the network.

In summary, both UNet and Residual UNet are popular architectures for image segmentation, with Residual UNet being an improvement that incorporates residual connections to handle deeper networks and achieve better performance in certain scenarios.

2.4 Related Work

In recent times, there has been a significant increase in the utilization of neural network methods. Specifically, the field of brain tumor segmentation has witnessed the development of a wide array of techniques. These techniques encompass traditional approaches such as region growing and active contours. Moreover, advanced deep learning techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have also been applied in this domain.

In the field of computer vision and image processing, convolutional neural network (CNN) (Anaraki, A. K., Ayati, M., & Kazemi, F., 2019) has always occupied the mainstream position. Researchers have lately begun to use deep convolutional neural networks' feature learning

capabilities for structured prediction challenges like segmentation. It has excelled at classification, segmentation, and object detection, along with many other computer vision tasks. The overall number of CNN-based approaches has grown recently, showcasing their effectiveness in segmenting brain tumors. The approaches can be different from one another by the input dimensions and size, filter size, network depth, connection paths, and other factors. However, the majority of medical image segmentation techniques, including the brain tumor segmentation, utilize Fully Convolutional Networks and Unet architectures.

Due to the intricate nature and crucial importance of the task, numerous research papers have been published on Brain Tumor Segmentation. To address this limitation, another research paper by (Triplanar Ensemble of 3D-to-2D CNNs with Label-Uncertainty for Brain Tumor Segmentation, 2020) proposed a method involving the use of 2D networks in axial, sagittal, and coronal views, followed by the fusion of their outcomes to incorporate 3D features. However, this approach utilized features from cross-planes instead of the complete 3D space, which made it challenging to employ prior shape and position information for robust brain tumor segmentation. Consequently, this approach might restrict the segmentation performance in practical scenarios.

Paper (Caliva, F., Iriondo, C., Martinez, A. M., Majumdar, S., & Pedoia, V, 2019) designed a CNN for 3D segmentation that employs a Hough voting technique. Nevertheless, their technique is not end-to-end and only works for compact blob-like structures.

In our research, we introduce a deep learning-based method for accurate bladder segmentation. Our approach combines a convolutional neural network (CNN) with a 3D fully connected conditional random fields recurrent neural network (CRF-RNN). Additionally, we propose a novel preprocessing technique called dual-channel preprocessing to enhance the segmentation performance of our approach.

Despite its advantages, CRF-RNN comes with some drawbacks. It involves a high computational cost during inference, and its training process is complex. Additionally, RNNs are designed to capture temporal dependencies in sequences, but for brain tumor segmentation, spatial relationships are more important than temporal ones. The use of recurrent neural networks (RNNs) can incur high computational costs and necessitate a substantial amount of data for effective

training. This can pose limitations in medical imaging tasks, where the availability of data is often limited in terms of size and quality.

Unet is a medical image segmentation architecture that uses a symmetric encoder-decoder network with skip links to improve details. Encoder-decoder networks are a specialized form of deep learning architecture designed to address segmentation challenges effectively. These networks excel at capturing essential features by encoding the input image into a condensed representation. Subsequently, they decode this representation to produce a segmented mask, thus facilitating accurate segmentation of objects within the image. These methods have been utilized in brain tumor segmentation to improve the accuracy and efficiency of the segmentation process. Various versions of Unet, such as Unet++, have been employed for this purpose (Xu, Dan and Zhou, Xidong and Niu, Xuefen and Wang, Junwei, 2020), and ResUnet (Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., & Mehmood, Z., 2020), increase image segmentation performance even more.

The Unet architecture and its variations exhibit exceptional performance in segmenting multi-class brain tumors. Although deep learning techniques are generally effective at representing features, the limited receptive field of convolution kernels presents difficulties in capturing explicit global contexts. This limitation hinders the learning of global semantic information, which is vital for dense prediction tasks such as segmentation. One possible solution is to train models with different depths independently and then combine the results. However, deploying networks with varying decoders introduces its own set of challenges.

The first attempt at utilizing the Transformer in 3D (Wang, W., Chen, C., Ding, M., Yu, H., Zha, S., & Li, J., 2021) Convolutional Neural Networks (CNN) for 3D MRI brain tumor segmentation was proposed. In their method, they introduced an encoder-decoder architecture, where the encoder network utilizes 3D CNN to extract spatial features from the volumetric MRI data. By downsampling the input 3D images, the encoder generates compressed volumetric feature maps that effectively capture local 3D context information. Subsequently, a multimodal brain tumor segmentation is performed for each of these feature maps.

To enable global context modeling, the authors transform the image into a vector, which is then fed into the Transformer. The Transformer is known for its capability to model long-range

dependencies efficiently. By employing this approach, the model can capture important contextual information across the entire image.

On the other hand, the 3D CNN decoder uses the feature embedding obtained from the Transformer and applies a progressive up-sampling technique to generate a complete resolution segmentation map prediction. This allows the model to reconstruct the segmentation results based on the encoded global information, resulting in accurate and detailed segmentation maps for brain tumor regions.

A paper introduced a brain tumor segmentation approach that combines a transformer (Wang, W., Xie, E., Li, X., Fan, D. P., Song, K., Liang, D., ... & Shao, L. , 2022) with a parallel convolutional network. Their proposed model achieved impressive results, obtaining the highest average Dice scores of 83.72% and 86.32% on the BraTS2019 and BraTS2018 datasets, respectively.

2.4.1 Summary of Related works

In recent years, the field of brain tumor segmentation has witnessed a surge in neural network approaches, ranging from traditional methodologies to advanced deep learning techniques. Convolutional Neural Networks (CNNs) have played a central role in computer vision and image processing, excelling in tasks such as classification, segmentation, and object detection. Many CNN-based approaches have demonstrated success in brain tumor segmentation, utilizing various architectures and configurations. Despite these advancements, challenges remain, and researchers have explored novel approaches such as 3D CNNs, Transformer-based models, and Unet architectures to improve segmentation accuracy. However, each method comes with its own advantages and limitations. This summary introduces a variety of literature that explores the application of these techniques in brain tumor segmentation, highlighting their strengths and potential constraints.

Table 2-1 summary of related works

Architecture	Description	Citation	Advantages	Limitations
Unet	- Convolutional neural network architecture designed for biomedical image segmentation.	(Ronneberger et al., 2015)	- U-shaped architecture provides an efficient feature extraction and upsampling process.	- May struggle with capturing explicit long-range dependencies due to limited receptive fields.

Architecture	Description	Citation	Advantages	Limitations
EEA Unet	- "Enhanced Encoder Atrous Unet" derived from UNet for retinal vessels segmentation.	(Sathananthavathi V., Indumathi G., 2021)	- Improves thin vessel extraction for retinal vessel segmentation.	- May still face challenges in segmenting certain complex structures.
ResNet-UNet	- Combines the traditional UNet structure with ResNet-34 for image segmentation.	(Vivekraj A., Sumathi S., 2023)	- Benefits from both UNet's segmentation capabilities and ResNet's deep learning and residual connections.	- May encounter difficulties in certain segmentation tasks that require precise localization.
Residual unet	- Extension of UNet that incorporates residual connections from ResNet.	(Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., & Mehmood, Z., 2020)	- Handles deeper architectures effectively and achieves better performance.	- Increased depth and complexity may lead to higher computational costs during training and inference.
Res-UNet-FSOA	- Residual unet-based architecture for cranial nerve segmentation.	(Vivekraj A., Sumathi S., 2023)	- Incorporates Fractional Calculus (FC) and Snake Optimizer (SO) for optimization.	- May require careful tuning of hyperparameters for optimal performance, and the use of advanced optimization techniques can increase the complexity of the training process.
2D Networks with Fusion	- Uses 2D networks in axial, sagittal, and coronal views, followed	by fusion to incorporate 3D features.	- Incorporates 3D features through the fusion of 2D networks in multiple views.	- May still face challenges in fully incorporating 3D spatial relationships for segmentation
Hough Voting CNN	- CNN for 3D segmentation with a Houghvoting technique.	(Caliva F., Iriondo C., Martinez A. M., Majumdar S., & Pedoia V, 2019)	- Suitable for segmentation of compact blob-like structures.	- Not an end-to-end technique and may not be as effective for more complex segmentation tasks.

Architecture	Description	Citation	Advantages	Limitations
CNN + CRF-RNN	- Utilizes a CNN and a 3D fully connected conditional random fields recurrent neural network (CRF-RNN) for accurate bladder segmentation.	Caliva, F., Iriondo, C., Martinez, A. M., Majumdar, S., & Pedoia, V, 2019)	- Utilizes a CRF-RNN to capture spatial relationships and improve segmentation performance.	- CRF-RNN may be computationally expensive during inference, and RNNs may require a large amount of data for training, which can be limiting in medical imaging tasks with limited data availability.
Transformer in 3D	- Introduces an encoder-decoder architecture where the encoder uses 3D CNN for spatial feature extraction, and the image is transformed into a vector and processed by the Transformer to capture global contextual information.	(Wang W., Chen C., Ding M., Yu H., Zha S., & Li J., 2021)	- Efficiently captures global contextual information and long-range dependencies through the Transformer's self-attention mechanism.	- Transformer's training process can be complex and computationally expensive.
Transformer-Based Model	- Uses Transformers' self-attention mechanism for brain tumor segmentation.	(Chen J., Liang C., Wang T., Shi F., Wang X., & Shen D., 2021)	- Dynamically highlights key features and captures both low-level and high-level information through self-attention.	- Transformer-based models may require substantial computational resources during training and inference.
Parallel Convolutional Network	Combines a Transformer with a parallel convolutional network for brain tumor segmentation.	(Wang W., Xie E., Li X., Fan D. P., Song K., Liang D., ... & Shao L., 2022)	- Impressive results in brain tumor segmentation with high average Dice scores.	- The approach may still require further evaluation on large and diverse datasets to assess its generalizability and robustness.

2.5 Research Gap

Brain tumor detection and segmentation are widely used techniques in brain magnetic resonance imaging (MRI) analysis for quantifying and visually representing the anatomical structures of the brain, detecting pathological regions, planning medical procedures, and guiding treatments. Due to the wide variety in the size, shape, location, and appearance of brain tumors, brain tumor detection and segmentation are challenging tasks that need extensive experimental investigation. Deep Neural Network algorithms like UNet outperform many semantic detection and segmentation tasks. However, UNet's shallow networks limit the model's ability to extract abstract and global contextual features.

The shallow network structure of U-Net can be a limitation in scenarios where capturing long-range dependencies and global context is crucial. In such cases, the model may struggle to learn complex relationships between distant image regions or capture global patterns that are essential for accurate segmentation.

3. CHAPTER THREE: METHODOLOGY

In this chapter provides a detailed explanation of the proposed method's architecture and experimental setup. It covers the dataset used for this work, as well as the data preprocessing techniques employed. Furthermore, the model training process is briefly explained. Inquiring the proposed system understanding way of development method, design the proposed solution, describe the testing and implementation method and tools that used to achieve the proposed solution.

3.1. Overview of the Proposed Methods

As shown in the Figure 3-1, the overall research process started from area definition from broader literature and we narrowed down to specific to deep learning-based brain tumor detection and segmentation related works. The input dataset was preprocessed via patching and minmax normalization and given to detection and segmentation backbone. We further refined detection and segmentation backbone to residual UNET for the tumor detection and segmentation in which the method proposed has the ability to leverage both local and global contextual features for improved region of interest prediction. For the effective proposed method evaluation we employed, mean over Union, dice similarity, accuracy, sensitivity, recall, and various other qualitative metrics.

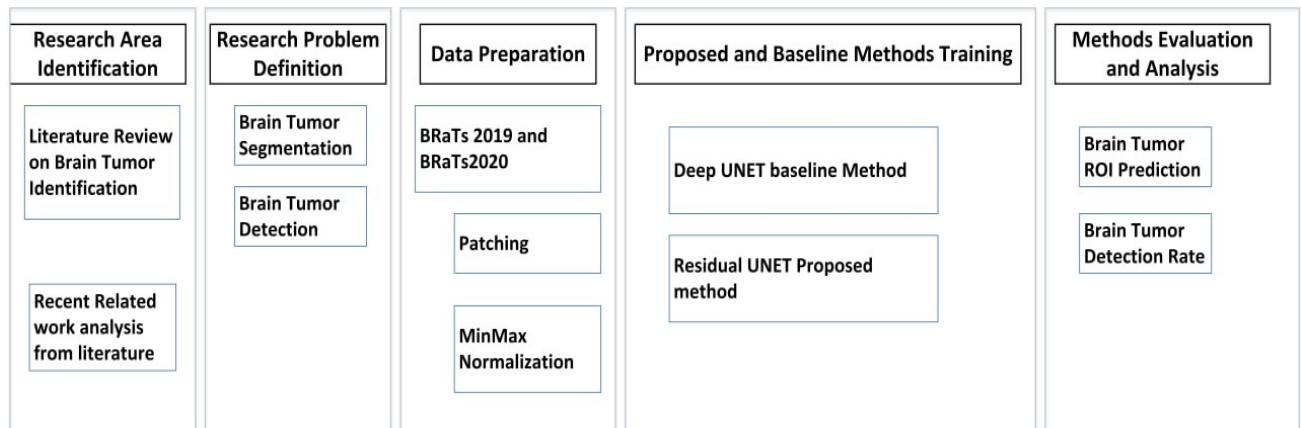


Figure 3-1Generic research process workflow

3.2. Dataset

The process of data collection entails systematically gathering and measuring relevant information on specific variables of interest with the aim of effectively addressing research inquiries. The

primary goal of data collection is to create suitable training and testing datasets that are essential for developing the desired solution. In our case collect the dataset, provided by downloading image from different website¹, ² and ³.

In deep learning data is the most major important thing that is used to develop the model. In deeplearning huge number of datasets are needed for training. In this study two types of datasets are used for implementing the model which is the training set and testing set.

3.3. Development and Implementation Tools

There are materials that are needed to make or distribute this work or research, and they are divided into two parts hardware tools and software tools.

3.3.1. Hardware tools

To properly implement this investigation, we require hardware tools. In the table below, you'll find a list of the hardware tools required for this investigation.

Table 3-1 Hardware components

No	Device name	Roll in the study
1	GPU	Google Colab with 12G graphic card
2	Laptop	Core i5 2.28Ghz, 8GB RAM, 1TB Hard disk for document processing and result generation
3	Hard disk	To store large amount of data sets prepared for training

3.3.2. Software tools

Different software and coding tools will be utilized to implement the research through coding, and writing, and these are given below with their descriptions.

¹<https://www.kaggle.com/datasets/aryashah2k/brain-tumor-segmentation-brats-2019>

²<https://www.kaggle.com/datasets/awsaf49/brats20-dataset-training-validation>

³<https://www.kaggle.com/datasets/dschettler8845/brats-2021-task1>

Microsoft office packages: - It is used to prepare the study document and create slides for presentation, and Visio is used to create the proposed system's diagrams and flowcharts.

Mendeley Desktop: - For its reference and bibliography management.

Anaconda: For generating training weight, testing result and other visualizations.

Keras with TensorFlow as back end: - was used for training the proposed and baseline model

Jupyter Notebook: generating different results such as confusion matrix, segmentation results, train-loss curve plotting.

Scikit-learn: In Python, there exists a comprehensive library that encompasses a diverse selection of unsupervised and supervised learning algorithms. Leveraging well-known technologies like NumPy, pandas, and Matplotlib, this library offers a broad array of tools to facilitate various machine learning tasks.

3.4. Baseline Model

3.4.1. DeepUnet

Deep U-Net is its increased depth compared to U-Net. Deep U-Net incorporates additional convolutional layers in both the encoder and decoder paths, allowing for the extraction of more abstract and complex features. The deeper architecture helps in capturing larger context and global information, improving the model's ability to understand the image and perform accurate segmentation.

The baseline neural network architecture consists of an encoder network (also called a contraction path) and a decoder network (expanding path). The encoder network comprises five convolutional blocks, while the decoder network has four convolutional blocks. Each block in both networks comprises two convolutional layers, followed by batch normalization (BN) and a rectified linear unit (ReLU). The use of ReLU activation function and BN is essential for improving the network's learning speed and stability. Additionally, BN helps to normalize the inputs to each layer and reduce the internal covariate shift.

To improve the precision of segmentation, the baseline neural network architecture utilizes different kernel sizes in the encoder and decoder networks to capture various features.

Deep neural networks are effective in learning hierarchical representations of data as each layer transforms the input data into a higher-level representation that can be further transformed by subsequent layers. This allows the network to capture complex dependencies and relationships between the input data, resulting in better performance on complex tasks.

In the case of brain tumor segmentation, deep neural networks can identify the critical features of tumor regions and differentiate them from non-tumor regions by learning shape, texture, and intensity information from brain images. By extracting these features hierarchically, with each layer building on the features learned by the previous layer, the network can capture complex relationships and dependencies between the input data, leading to more precise segmentation results. This can significantly enhance the accuracy and efficiency of automated brain tumor segmentation, which is crucial for medical treatment planning and diagnosis.

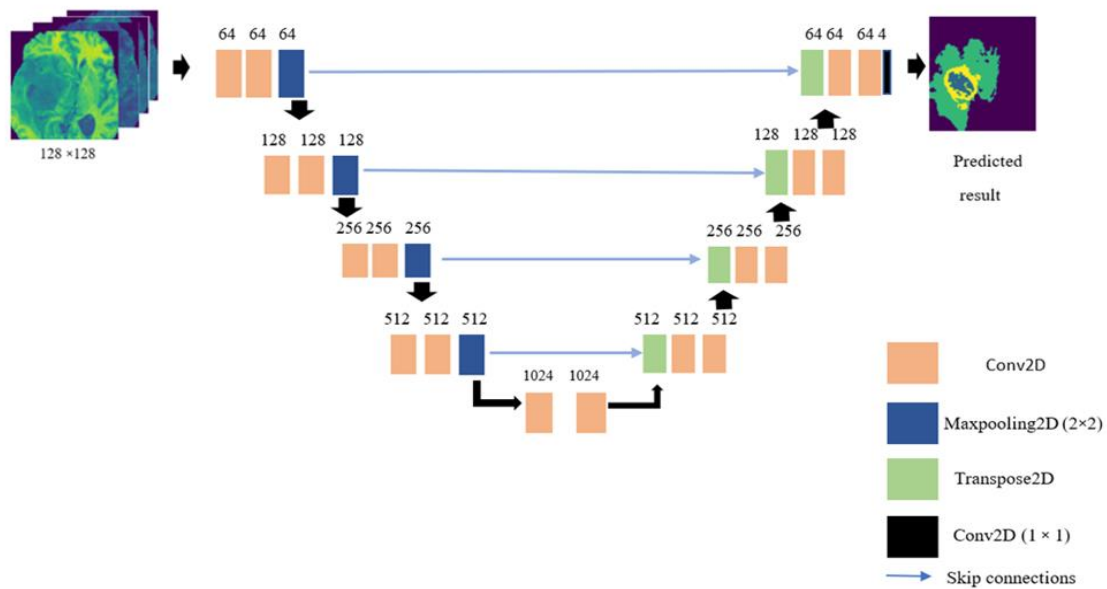


Figure 3-2The Baseline DeepUnet Architecture

4. CHAPTER FOUR: PROPOSED ARCHETICTURE

The process of training and validating brain images for predicting the region of interest (ROI) involves several stages. In the preprocessing stage, brain images are acquired, including both tumor and non-tumor images, and then normalized to ensure consistent pixel intensity ranges. Additionally, image transformations like resizing and cropping are performed to standardize the images. Moving on to the training and validation stage, an encoder-decoder model architecture is constructed. The encoder extracts relevant features, while the decoder reconstructs the images using these features. The preprocessed dataset is split into training and validation sets, with the former used for model training and the latter for evaluating performance and fine-tuning hyperparameters. During model training, the encoder-decoder model learns to encode images into a latent space representation and decode them to reconstruct the original images. Model evaluation involves assessing metrics such as accuracy and loss using the validation set. In the testing stage, a separate set of brain images is preprocessed using the same techniques as before. These preprocessed test images are then fed into the trained encoder-decoder model, which encodes them into a latent space and generates the predicted ROI by decoding them. This comprehensive process enables the analysis and detection of tumors or other regions of interest in brain images.

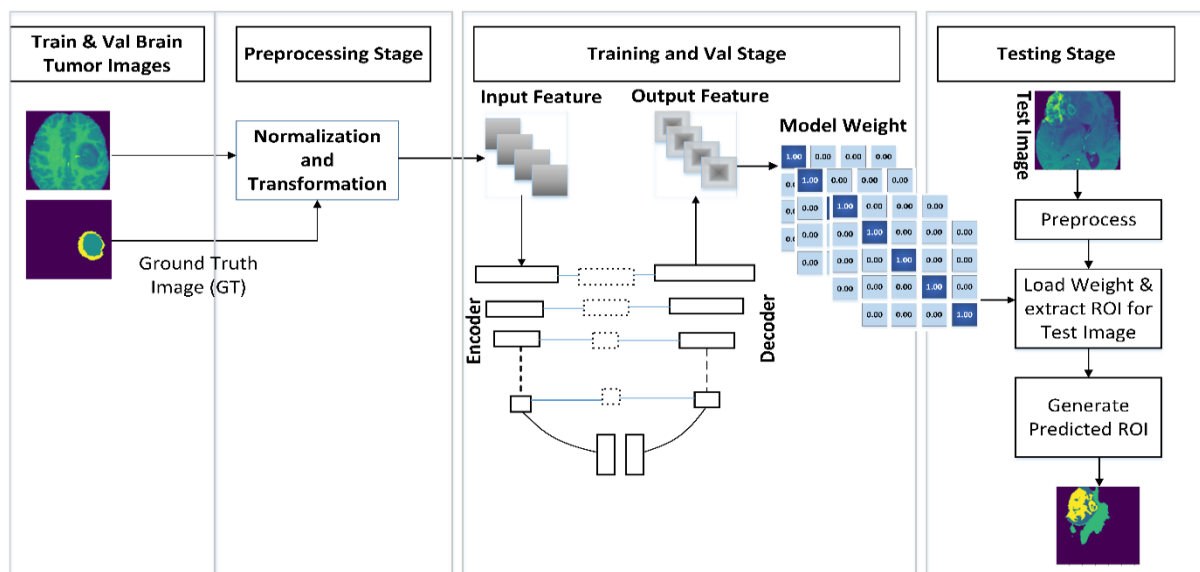


Figure 4-1 Overview of the Methodology

4.1 Residual Unet

The proposed model consists of an encoder and decoder network, comprising ten and four convolutional blocks, respectively. Each convolutional block consisting of two convolutional layers followed by a ReLU activation function and batch normalization (BN). Maxpooling and dropout functions are applied after every two blocks of the encoder network. In addition, the proposed model has a residual skip connection block to extract both local and global contexts.

The encoder network has a deep layer that can extract detailed information effectively, and the decoder network has a residual block with both small and larger receptive fields. The residual blocks allow the decoder to restore features completely, improving the overall performance of the network. The following image illustrates the architecture of improved UNet models.

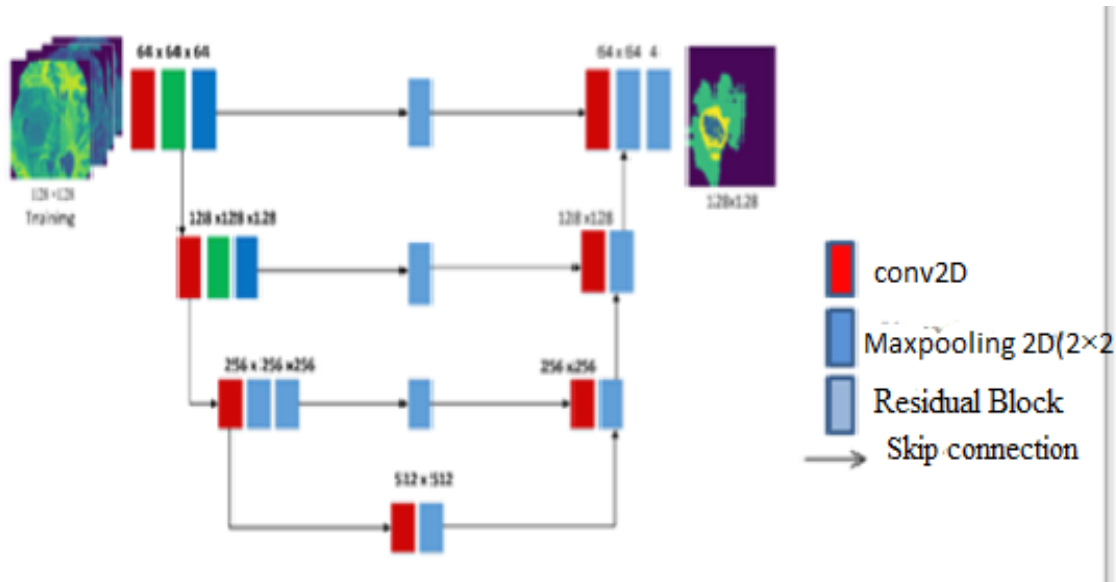


Figure 4-2 The Proposed Residual UNet Architecture

Expanding the encoder network of an image segmentation model can enhance its capacity to detect and depict intricate features and patterns in the input image by increasing the network's depth enables it to acquire more complicated and abstract representations of the input image, which can aid in distinguishing between various classes or regions in the image as it shown fig 4.2. Additionally, incorporating more layers into the network can prevent over fitting, which arises when the network memorizes the training data instead of generalizing to new data. By learning more comprehensive and resilient representations of the input data, the network can reduce over fitting and consequently improve its ability to generalize to new data.

The decoder network, on the other hand, is composed of a series of up sampling layers (Conv2DTranspose) and four convolutional blocks which are responsible for reconstructing the segmentation mask from the low dimensional high-level features learned by the encoder. Conv2DTranspose uses 3×3 filters to learn how to up sample the input feature map while preserving spatial information. Then the result obtained from the Conv2DTranspose layer is combined with the output produced by the corresponding residual skip connection block. The residual module involves connecting four parallel convolutional layers that are responsible for processing the input data independently, and the outputs of each network are combined to produce a final output.

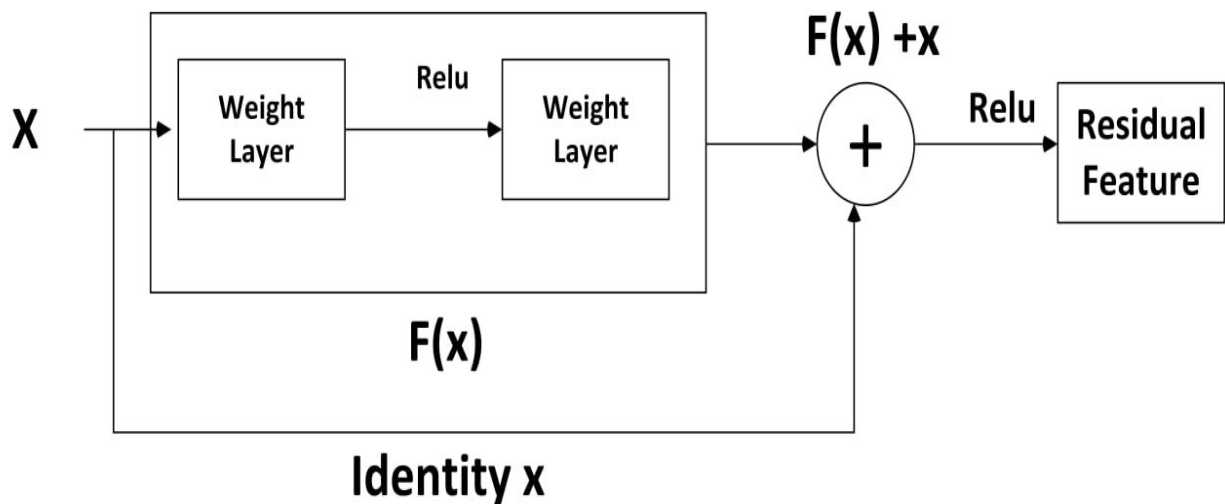


Figure 4-3 Shows the architecture of skip connection module

To address the issue of vanishing or exploding gradients, this architecture introduced a concept known as Residual Blocks. Within this network, skip connections are employed as a technique. These connections link the activations of a layer to subsequent layers by bypassing certain intermediate layers, thereby forming a residual block.

ResidualUnets are constructed by stacking these residual blocks together. The approach behind this network is instead of layers learning the underlying mapping, we allow the network to fit the residual mapping.

By incorporating skip connections of this nature, the architecture gains the benefit of being able to bypass any layers that may negatively impact its performance, thus mitigating potential issues through regularization. As a result, this enables the training of exceptionally deep neural networks without being hindered by concerns related to vanishing or exploding gradients.

Finally, the decoder network integrates the high-level and low-level features obtained from the combined feature maps, and the output is passed through a ReLu activation function, which produces a probability map indicating the presence or absence of the object of interest in each pixel of the input image.

Residual networks solve the problem of vanishing gradients and allow the process of training neural networks with a significant depth by using skip connections to add the output of previous layers to the current layer, residual networks allow the gradient to flow directly to earlier layers, avoiding the problem of vanishing gradients. In general, the strengths of Unet architectures can be improved by combining UNet and residual networks. The skip connections in residual networks enable the UNet to capture more high-level features and learn more robust representations. This can lead to better segmentation performance, especially in tasks with complex and diverse structures. Additionally, residual networks can help prevent over fitting, allowing for more effective training and better generalization performance.

4.2 Dataset and Its Preprocessing

Since 2012, the Brain Tumor Segmentation (BraTS) challenge has been focused on developing effective strategies for the task of identifying and segmenting brain tumors observed in magnetic resonance imaging (MRI) data collected from multiple imaging modalities. Although BraTS has expanded its scope to include patient overall survival prediction, our research has focused solely on tumor region segmentation. Two publicly available datasets, BRATS 2019 and BRATS 2020, are used to train our proposed model. The training sets of BRATS2019 consist of 3240 MRI Images.

The BRATS2020 dataset contains 2340 training sets and 250 validation sets, and 650 images set for testing. with both datasets containing expert-labeled annotations for each patient along with MRI scans of T1, T1 post-contrast (T1CE), T2, and T2 Fluid Attenuated Inversion Recovery (Flair) volumes. While the validation sets of both BRATS 2019 and BRATS2020 contain 3D MRI

volumes of 125 patients of unknown grade, the training sets were used in this study since their ground truth data is provided. Due to memory constraints, only 50 samples are selected from each dataset to train the proposed methods. An example image from the BRATS2020 dataset is shown in the following image.

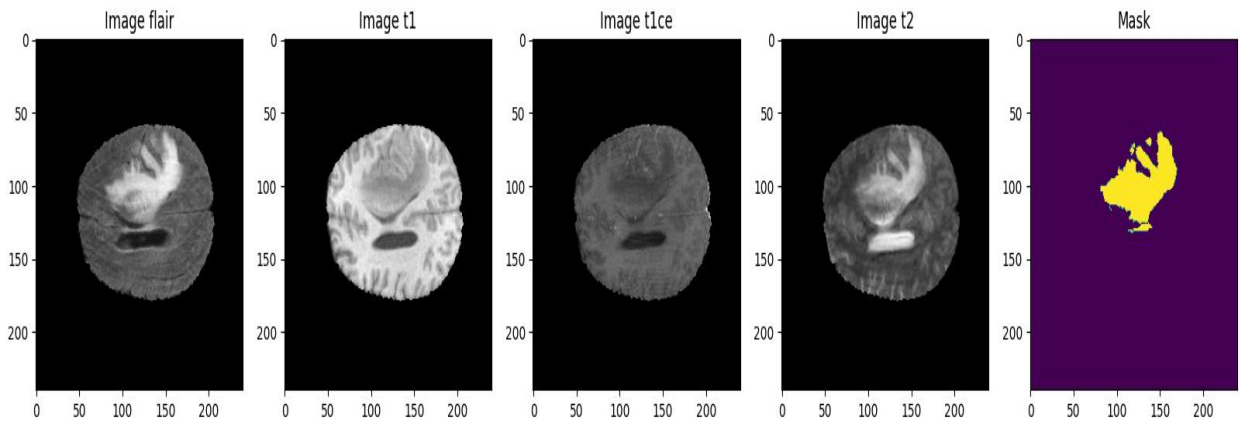


Figure 4-4 Sample Image from BRATS2020

Initially, the BraTS segmentation project concentrated on identifying four distinct glioma sub-regions in MRI images, which were defined as the necrotic region (NCR), edema (ED), non-enhancing tumor (NET), and enhancing tumor (ET) and labelled as 1, 2, 3, and 4, respectively (Sun, L., Zhang, S., Chen, H., & Luo, L., 2019). However, in certain cases, the four regions are merged into three labels.

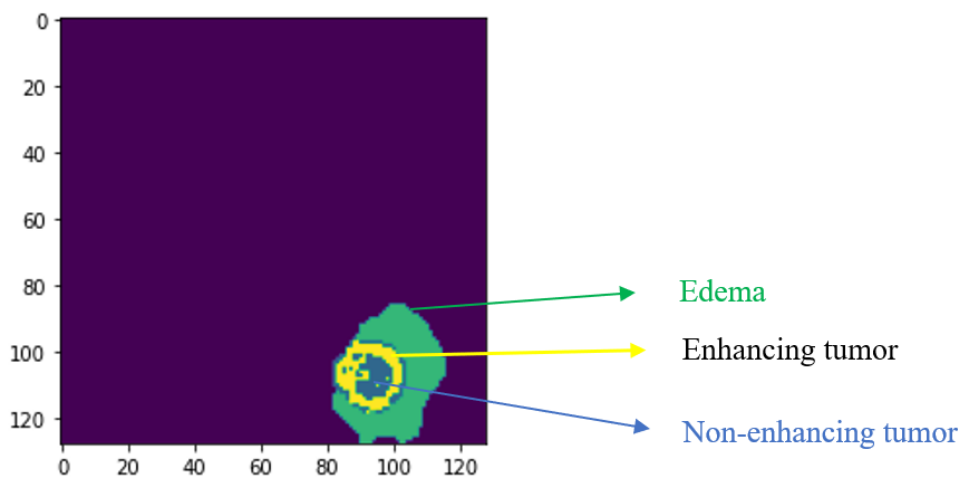


Figure 4-5 Sample Label from BRATS2020

Each pixel in the image is assigned one of the following labels: label 1 for necrotic and non-enhancing tumor core, label 2 for peritumoral edema, and label 3 for enhancing tumor.

A brain image should first go through a number of preprocessing procedures before being used for further research. Preprocessing is crucial as it makes the raw data appropriate for machine learning models and enhances the performance of the models. The two essential steps involved in this thesis are normalization and patching.

4.2.1 Normalization

Normalization involves scaling the intensity values of the brain images so that they fall within a specific range. This is done to reduce the effect of inter-subject variability and to make the data consistent across different subjects. Normalization is achieved by applying min-max scaling.

Deep learning methods generally perform better when the input features are on a similar scale and close to normally distributed. The saying “Garbage in, garbage out” is often used in machine learning, indicating that if the input contains low-quality data, the model’s performance will suffer. Therefore, preprocessing is a crucial step to obtain high-quality data and improve the model’s performance. Magnetic resonance images are acquired differently and have different content and contrast, thus requiring standardization and normalization of the input images.

In proposed method, variables measured at different scales may not have equal importance in model fitting and learning, potentially leading to bias. To address this issue, feature-wise standardization, such as min-max Scaling, is typically utilized before fitting the model. In this task, the input data is scaled by the min-max scaling method.

MinMax scaling (also known as min-max normalization) is a type of feature scaling, where the features are rescaled so that they have the same range of values. In this scaling, the minimum value of the feature is mapped to 0, and the maximum value is mapped to 1. The other values are then scaled proportionally in between the minimum and maximum values. For example, suppose a feature with values ranging from 0 to 256 is given. After applying min-max scaling, the minimum value up to 128 would be mapped to 0, and the maximum value of up to 256 would be mapped to 1. A value of 150 would then be mapped to $(150-100)/(200-100) = 0.5$.

The min-max scaling is given as follows.

$$\hat{X}_{ij} = \begin{cases} \frac{X_{ij} - X_{\min}}{X_{\max} - X_{\min}} , & X_{ij} \neq 0 \\ 0 , & \text{else} \end{cases} \dots \dots \dots 3.1$$

Where $\hat{X}_{ij}$ is scaled value at ij^{th} position in which $i, j \in [0,239]$. $X_{\max}$ and $X_{\min}$ indicates the maximum and minimum values for all X_{ij} respectively. Because the shape of the distribution remains constant, the scaled value has the same proportion as the original data. MinMax scaling is often used in machine learning to improve the performance and stability of algorithms, particularly when dealing with features that have vastly different ranges of values.

4.2.2 Patching

After the data is Transformed into a given range, the next step is combining modalities and patching. Patching in machine learning is a technique used to divide a large image into smaller sub-images or patches. These patches are then used as input to a machine learning algorithm for training or inference.

When working with large images, patching can help reduce the memory requirements of the algorithm. Instead of loading the entire image into memory, patches can be loaded and processed sequentially, thus reducing memory usage. Since the images used in this task are too large to fit on a GPU all at once, it typically should be broken into smaller patches for training the deep learning model. As illustrated in the following figure, the original images have dimensions of $240 \times 240 \times 155$, resulting in a total of 8,928,000 pixels. This large size requires significant computational resources and can also cause data imbalance due to the prevalence of background pixels.

Patching is employed to remove unnecessary background information and make the data more manageable. This can help improve the accuracy and robustness of the model. Reducing the size of an image patch is an effective way to speed up the model training process, but it may also result in information loss. In our study, this trade-off is observed and decided to crop the images into a medium size of 128×128 pixels, which still provides a reasonable amount of information. In general, the patching technique is a valuable approach in machine learning that can enhance the performance and efficiency of models when dealing with extensive or intricate images.

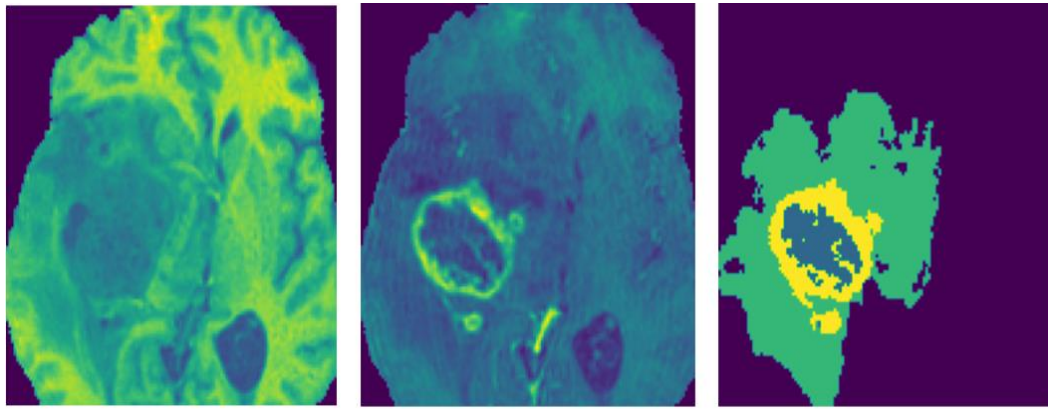


Figure 4-6 image Patches

Subsequently, the patches are inputted into the network, where each network layer corresponds to a multi-channel image with spatial coordinates (2D information). In the neural network, each layer corresponds to a multi-channel image with spatial coordinates. The network is trained using the training data, while the validation data is used to monitor the training process and prevent overfitting. Finally, the performance of the model is evaluated using the testing data, which was not used during the training process.

To partition the dataset, a standard split of 80% for training and validation out of which (90% for training data and 10% for validation) and 20% of the training data reserved for testing purposes. By following this approach, the model receives training on a dataset of considerable size, thereby preventing overfitting. The visualization below illustrates the distribution of data in this study.

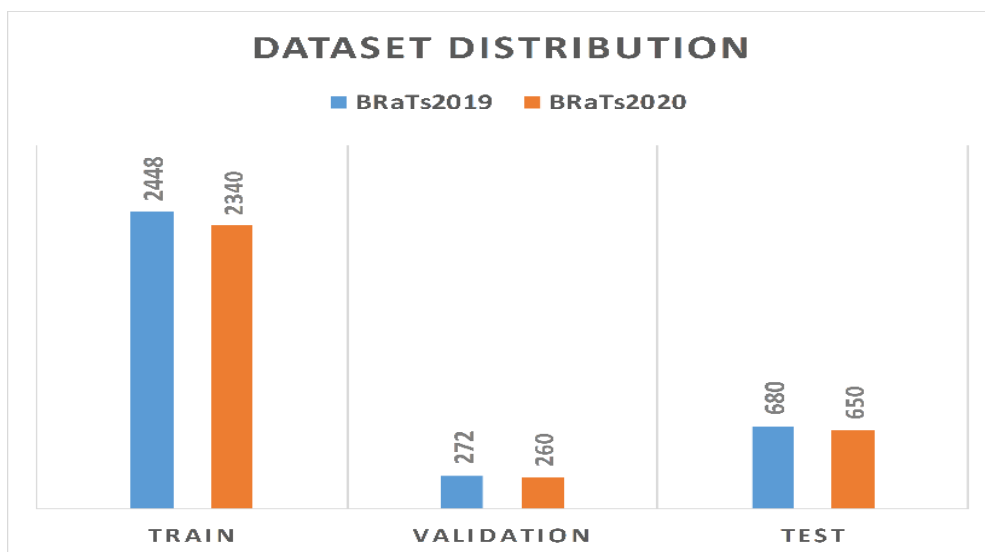


Figure 4-7 Data Distribution

4.3 Training the Models

The experiments are conducted to evaluate the performance of our proposed model on a dataset of 50 volumes patched into 3400 for 3250 MRI images for each dataset. The proposed model was implemented using Python and the Tensor Flow framework. The experiments were performed on a machine utilizing Google Colab. The architecture takes an image with a shape of $128 \times 128 \times 3$, and trained in 80 epochs with a batch size of 8.

4.3.1 Optimizer

The proposed and the baseline models are optimized through an Adams optimizer. The Adam optimizer also includes a bias correction mechanism that compensates for the fact that the estimates of the first and second moments may be biased towards zero in the initial phases of the training. This bias correction is important during the initial phases of training when the gradient estimates are noisy. The adaptive learning rate and bias correction mechanisms of Adam optimizer make it an effective optimization algorithm for deep learning models.

4.3.2 Loss function

This study used a popular loss function, Categorical cross-entropy (CCE) loss, is used in to train the models for multi-class classification tasks. It measures the dissimilarity between the predicted probability distribution and the true probability distribution of the classes. In the case of categorical cross-entropy loss, the predicted probability distribution is obtained by passing the input through a MinMax function, which transforms the output of the model into a probability distribution over the classes. The true label is represented as a one-hot encoded vector, where the value of 1 is assigned to the true class and 0 to all other classes.

The cross-entropy loss is then calculated as the negative log-likelihood of the true class under the predicted probability distribution. In other words, the loss is a measure of how well the model can predict the true class, with lower values indicating better performance. Mathematically, it can be calculated as follow.

The formula for Categorical Cross-Entropy (CCE) loss is:

$$CCE = \frac{-1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij}) \dots \dots \dots 3.2$$

where N is the number of samples, M is the number of classes, y_{ij} and p_{ij} are the true and the predicted labels of sample i for class j.

A popular loss function, Dice loss, also known as Sorensen–Dice coefficient is used in this work. The Dice loss is a commonly used loss function in image segmentation tasks, including brain tumor segmentation. In brain tumor segmentation, The Dice loss is a measurement of the difference between the predicted segmentation mask and the actual segmentation mask. The predicted mask represents the tumor region that the model predicts, while the ground truth mask is the actual tumor region as annotated by a human expert. The Dice loss function is defined as:

$$\text{Dice loss} = 1 - (2 * \text{intersection}) / (\text{predicted mask} + \text{ground truth mask}) \dots \dots 3.3$$

where the intersection is the element-wise multiplication between the predicted mask and the ground truth mask, and the sum of predicted mask and ground truth mask gives the total number of pixels. Thus, the goal of training a segmentation model using the Dice loss is to minimize its value, which would correspond to maximizing the similarity between the predicted and ground truth masks.

4.3.3 Regularization

To improve the performance of the model and prevent over fitting, an early stopping regularization technique have been implemented during training. It involves monitoring the performance of the model on a validation set during training and stopping the training process when the performance on the validation set stops improving. This is done to prevent the model from continuing to learn from the training set and instead generalizing well to new, unseen data. Specifically, a patience value of 10 iterations on losses was set, which means that if there is no improvement in the validation loss after 10 epochs, the training process will be stopped early.

The idea behind it is that the model's performance on the training set will continue to improve as it learns to fit the training data more and more closely. However, at some point, the model may begin to overfit the training data and start to perform poorly on new, unseen data. During the training process, it's possible to detect when a model has started to overfit by monitoring its

performance on a validation set. By doing so, the training process can be stopped before the overfitting becomes too severe.

To further enhance the regularization of the model, a dropout of 0.3 was applied after each convolutional block during training. This technique helps to prevent over fitting by randomly dropping out 30% of the neurons during each training iteration, which encourages the model to learn more robust features. The dropped-out neurons temporarily ignored in the calculation of the outputs of that iteration. This encourages the model to learn more robust features by preventing it from relying too heavily on any one set of neurons. By randomly dropping out neurons during training, the model becomes more resilient to noise and variations in the input data.

4.4 Detection and Segmentation Evaluation Matrices

Metrics allow to measure the prediction performance of machine learning model. The most used metrics used to evaluate how well any segmentation methods include Dice Similarity Score, Mean IOU, accuracy, specificity, sensitivity or recall, precision etc. These matrices evaluate neural network parameters by examining the values of the confusion matrix components, which provide a multi-faceted evaluation of the model's performance. The four components of the confusion matrix are True Positive, False Positive True Negative, and False Negative. To evaluate the model's performance in various contexts, these components are calculated based on each sample in the dataset, where 0 and 1 are used to represent negative and positive samples, respectively. Table 3-3 below provides a detailed description of each sample's classification based on the confusion matrix components.

Table 4-1 Evaluation matrix

Samples	Ground truth	Predicted labels
TP	1	1
TN	0	0
FP	0	1
FN	1	0

Accuracy: is the most widely used evaluation matrix in deep learning which measures the ratio of correct predictions to the whole samples (Müller, D., Soto-Rey, I., & Kramer, F, 2022).

Then the accuracy can be calculated as follow.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots \dots \dots 3.4$$

Specificity:is a metric commonly used in binary classification tasks, such as object detection, to evaluate the model's ability to correctly identify negative samples or background elements. In the context of object detection, specificity refers to the model's ability to accurately identify regions in an image that do not contain the objects of interest.

It indicates the amount of true negative that have been correctly identified and used to know the number of correct negative predictions (TN) over the whole actual negative elements (Wang, D. G., Brewster, J. D., Paul, M., & Tomasula, P. M. .. Molecules, ., 2015).

$$Specificity = \frac{TN}{TN + FP} \dots \dots \dots 3.5$$

Sensitivity (Recall):Recall measures the proportion of correctly detected objects out of all the ground truth objects. It is calculated as the ratio of true positives to the sum of true positives and false negatives (objects not detected by the model).

It indicates the number of true positives that have been correctly identified and it calculated as the ratio of the number of correct positive predictions over the whole actual positive elements (Fernandez-Moral et al., 2018).

$$Sensitivity = \frac{TP}{TP + FN} \dots \dots \dots 3.6$$

Precision:measures the proportion of correctly detected objects out of all the objects predicted by the model. It is calculated as the ratio of true positives (correctly detected objects) to the sum of true positives and false positives (objects incorrectly detected as positive).

It is used to calculate the number of correctly predicted positives over the total number of positive predictions (Vujovic, 2021).

$$Precision = \frac{TP}{TP + FP} \dots \dots \dots 3.7$$

DiceSimilarity Score (DSC): is a common metric for evaluating the effectiveness of image segmentation. DSC (Malhotra, R., Saini, B. S., & Gupta, S., 2022)measures the overlap between the predicted segment and the ground truth. It can be calculated through dividing is the area

covered by the intersecting between the ground truth and the predicted segment to the total area of union between two sets.

$$DSC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \dots\dots\dots 3.8$$

Where, A and B represents ground truth and predicted label. This metrics is mostly applied when the background pixels outweigh the foreground pixels.

Like Dice score, the mean Intersection over Union (mean IoU) is a commonly used evaluation metric for image segmentation tasks. It provides a single value to evaluate the segmentation model's overall performance by calculating the degree of similarity between the predicted segmentation mask and the ground truth mask for different classes. It can be calculated as follow.

$$IOU(A, B) = \frac{|A \cap B|}{|A + B|} \dots\dots\dots 3.9$$

Mean IoU: is calculated as the average of IoU scores for each class in the dataset. The main difference between mean IoU (Intersection over Union) and Dice score is the way they measure the similarity between the two masks. Mean IoU measures the proportion of intersection area between the predicted segmentation map and the ground truth map relative to their union. On the other hand, Dice score measures the overlap between two masks, which is the ratio of twice the area of intersection between the two maps to the total area of both maps.

4.5 Summary

This chapter provides an overview of the model architecture and experimental setup used in this task. The chapter starts by introducing the model architectures and the dataset used in this work as well as the preprocessing steps applied to the images to make them suitable for model training. The preprocessing steps include normalization and patching, which are crucial for obtaining high-quality data, reducing the size of the input images and eliminating unnecessary background. The chapter also provides a detailed explanation of the used for model training, including the learning rate, batch size, and number of epochs.

5. CHAPTER FIVE: IMPLEMENTATION OF THE PROPOSED ARCHITECTURE

5.1 Tools of Implementation

5.1.1 Hardware Tools

➤ Laptop:

- Operating System: Windows 10
- Processor: Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz, 2401 Mhz, 2 Core(s), 4 Logical Processor(s)
- Graphics: Intel(R) HD Graphics Family
- Installed memory (RAM): 8.00 GB
- System Type: 64-bit operating system, x64-based processor
- Google Colab

5.1.2 Software tools

Different software and coding tools will be utilized to implement the research through coding, and writing, and these are given below with their descriptions.

Anaconda: For generating training weight, testing result and other visualizations.

Keras with TensorFlow as back end: - was used for training the proposed and baseline model

Jupyter Notebook: generating different results such as confusion matrix, segmentation results, train-loss curve plotting.

Scikit-learn: In Python, there exists a comprehensive library that encompasses a diverse selection of unsupervised and supervised learning algorithms. Leveraging well-known technologies like NumPy, pandas, and Matplotlib, this library offers a broad array of tools to facilitate various machine learning tasks.

5.2 Training Procedure

To accomplish the overall objectives of this study, the following processes carried out.

- Dataset collection
- Data preprocessing
- Configuring Graphical Processing Unit (GPU) and upload the dataset
- Training and Testing

5.3 Proposed Architecture Implementation

5.3.1 The Encoder Implementation

The provided encoder decoder residual architecture consists of four blocks, each comprising convolutional layers, batch normalization with ReLU activation, max pooling, and dropout layers. It begins with Block 1, where the input image of size $128 \times 128 \times 4$ undergoes successive convolutions to produce a feature map of size $128 \times 128 \times 64$. Block 2 continues this process, reducing the spatial dimensions to $64 \times 64 \times 128$. Block 3 further reduces the dimensions to $32 \times 32 \times 256$, while Block 4 brings it down to $16 \times 16 \times 512$. Throughout the architecture, batch normalization and ReLU activation enhance the feature maps, and dropout layers help mitigate overfitting. These successive blocks enable the model to capture increasingly complex features while preserving spatial information. This encoder structure is typically followed by a decoder architecture for upsampling and reconstruction based on the encoded features.

Table 5-1 Encoder part of the architecture

Blocks	Layer type	Kernels	Input size
	Input Layer	-	$128 \times 128 \times 4$
1	conv2d_1	3×3	$128 \times 128 \times 64$
	conv2d_2	3×3	$128 \times 128 \times 64$
	BN +ReLU	-	$128 \times 128 \times 64$
	max_pooling2d	2×2	$64 \times 64 \times 64$
	Dropout	-	$64 \times 64 \times 64$
	Skip connection		$64 \times 64 \times 64$
2	conv2d_3	3×3	$64 \times 64 \times 128$
	conv2d_4	3×3	$64 \times 64 \times 128$
	BN +ReLU	-	$64 \times 64 \times 128$

	max_pooling2d	2×2	$32 \times 32 \times 128$
	Dropout	-	$32 \times 32 \times 128$
	Skip connection		$32 \times 32 \times 128$
3	conv2d_5	3×3	$32 \times 32 \times 256$
	conv2d_6	3×3	$32 \times 32 \times 256$
	BN +ReLU	-	$32 \times 32 \times 256$
	max_pooling2d	2×2	$16 \times 16 \times 256$
	Dropout	-	$16 \times 16 \times 256$
	Skip connection		$16 \times 16 \times 256$
4	conv2d_7	3×3	$16 \times 16 \times 512$
	conv2d_8	3×3	$16 \times 16 \times 512$
	BN +ReLU	-	$16 \times 16 \times 512$
	max_pooling2d	2×2	$8 \times 8 \times 512$
	Dropout	-	$8 \times 8 \times 512$
	Skip connection		$8 \times 8 \times 512$

Sample Code for the Encoder Architecture

```
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, ReLU,
MaxPooling2D, Dropout
from tensorflow.keras.models import Model
```

```
# Input layer
input_shape = (128, 128, 4)
inputs = Input(shape=input_shape)

# Block 1
conv1 = Conv2D(64, (3, 3), padding='same')(inputs)
conv1 = Conv2D(64, (3, 3), padding='same')(conv1)
conv1 = BatchNormalization()(conv1)
conv1 = ReLU()(conv1)
pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
drop1 = Dropout(0.5)(pool1)
skip1 = conv1

# Block 2
conv2 = Conv2D(128, (3, 3), padding='same')(drop1)
conv2 = Conv2D(128, (3, 3), padding='same')(conv2)
conv2 = BatchNormalization()(conv2)
conv2 = ReLU()(conv2)
pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
drop2 = Dropout(0.5)(pool2)
skip2 = conv2

# Block 3
conv3 = Conv2D(256, (3, 3), padding='same')(drop2)
conv3 = Conv2D(256, (3, 3), padding='same')(conv3)
conv3 = BatchNormalization()(conv3)
```

```

conv3 = ReLU() (conv3)
pool3 = MaxPooling2D(pool_size=(2, 2)) (conv3)
drop3 = Dropout(0.5) (pool3)
skip3 = conv3

# Block 4
conv4 = Conv2D(512, (3, 3), padding='same') (drop3)
conv4 = Conv2D(512, (3, 3), padding='same') (conv4)
conv4 = BatchNormalization() (conv4)
conv4 = ReLU() (conv4)
pool4 = MaxPooling2D(pool_size=(2, 2)) (conv4)
drop4 = Dropout(0.5) (pool4)
skip4 = conv4

# Create the model
encoder = Model(inputs=inputs, outputs=drop4)

```

5.3.2 The Decoder Implementation

The decoder architecture consists of four blocks that gradually upsample the input features to reconstruct the output. In Block 4, the input features are upsampled and concatenated with skip connection features, followed by dropout regularization and convolutional layers. The output is combined with the previous block's output through a skip connection. Block 3 further upsamples and concatenates features, applies dropout, and performs convolutional operations with batch normalization and ReLU activation. The output is again combined with the previous block's output. Block 2 follows a similar pattern but with different spatial dimensions. Finally, in Block 1, transposed convolution is used for upsampling, and features are concatenated, dropped out, and processed through convolutional layers. The output is passed through a 1x1 convolutional layer and an activation function to obtain the final output feature map. Again, a skip connection combines the outputs from the previous block and the Conv2D layer.

Table 5-2 Decoder Part of the Residual UNet Model

Blocks	Layer Type	Kernels	Input Size
4	UpSampling2D	2x2	8x8x512
	Concatenate	-	16x16x1024
	Dropout	-	16x16x1024
	Conv2D (conv2d_9)	3x3	16x16x512
	Conv2D (conv2d_10)	3x3	16x16x512
	BN + ReLU	-	16x16x512
	Skip connection		16x16x512
3	UpSampling2D	2x2	32x32x512
	Concatenate	-	64x64x640

	Dropout	-	64x64x640
	Conv2D (conv2d_11)	3x3	64x64x256
	Conv2D (conv2d_12)	3x3	64x64x256
	BN + ReLU	-	64x64x256
	Skip connection		64x64x256
2	UpSampling2D	2x2	128x128x256
	Concatenate	-	128x128x320
	Dropout	-	128x128x320
	Conv2D (conv2d_13)	3x3	128x128x128
	Conv2D (conv2d_14)	3x3	128x128x128
	BN + ReLU	-	128x128x128
	Skip connection		128x128x128
1			
	Conv2DTranspose	2x2	256x256x128
	Concatenate	-	256x256x192
	Dropout	-	256x256x192
	Conv2D (conv2d_15)	3x3	256x256x64
	Conv2D (conv2d_16)	3x3	256x256x64
	BN + ReLU	-	256x256x64
	Conv2D (conv2d_17)	1x1	256x256x1
	Activation ()	-	256x256x1
	Skip connection		256x256x1

Sample code for the decoder Architecture

```
from tensorflow.keras.layers import UpSampling2D, Concatenate, Dropout,
Conv2D, BatchNormalization, ReLU, Conv2DTranspose, Activation
from tensorflow.keras.models import Model
```

```
# Input layer
input_shape = (256, 256, 1)
inputs = Input(shape=input_shape)

# Block 4
up4 = UpSampling2D(size=(2, 2))(inputs)
concat4 = Concatenate()([up4, skip4])
drop4 = Dropout(0.5)(concat4)
conv9 = Conv2D(512, (3, 3), padding='same')(drop4)
conv10 = Conv2D(512, (3, 3), padding='same')(conv9)
conv10 = BatchNormalization()(conv10)
conv10 = ReLU()(conv10)
skip3_upsampled = UpSampling2D(size=(4, 4))(skip3)
skip4_resized = Conv2D(512, (1, 1))(skip4)
skip4_merged = Concatenate()([skip3_upsampled, skip4_resized])
skip_connection4 = Conv2D(512, (3, 3), padding='same')(skip4_merged)
conv10 = Add()([conv10, skip_connection4])

# Block 3
up3 = UpSampling2D(size=(2, 2))(conv10)
concat3 = Concatenate()([up3, skip3])
drop3 = Dropout(0.5)(concat3)
```

```

conv11 = Conv2D(256, (3, 3), padding='same')(drop3)
conv12 = Conv2D(256, (3, 3), padding='same')(conv11)
conv12 = BatchNormalization()(conv12)
conv12 = ReLU()(conv12)
skip2_upsampled = UpSampling2D(size=(2, 2))(skip2)
skip3_resized = Conv2D(256, (1, 1))(skip3)
skip3_merged = Concatenate()([skip2_upsampled, skip3_resized])
skip_connection3 = Conv2D(256, (3, 3), padding='same')(skip3_merged)
conv12 = Add()([conv12, skip_connection3])

# Block 2
up2 = UpSampling2D(size=(2, 2))(conv12)
concat2 = Concatenate()([up2, skip2])
drop2 = Dropout(0.5)(concat2)
conv13 = Conv2D(128, (3, 3), padding='same')(drop2)
conv14 = Conv2D(128, (3, 3), padding='same')(conv13)
conv14 = BatchNormalization()(conv14)
conv14 = ReLU()(conv14)
skip1_upsampled = UpSampling2D(size=(2, 2))(skip1)
skip2_resized = Conv2D(128, (1, 1))(skip2)
skip2_merged = Concatenate()([skip1_upsampled, skip2_resized])
skip_connection2 = Conv2D(128, (3, 3), padding='same')(skip2_merged)
conv14 = Add()([conv14, skip_connection2])

# Block 1
up1 = Conv2DTranspose(64, (2, 2), strides=(2, 2))(conv14)
concat1 = Concatenate()([up1, skip1])
drop1 = Dropout(0.5)(concat1)
conv15 = Conv2D(64, (3, 3), padding='same')(drop1)
conv16 = Conv2D(64, (3, 3), padding='same')(conv15)
conv16 = BatchNormalization()(conv16)
conv16 = ReLU()(conv16)
conv17 = Conv2D(1, (1, 1))(conv16)
output = Activation('sigmoid')(conv17)

# Create the model
decoder = Model(inputs=inputs, outputs=output)

```

This code snippet defines the decoder architecture with the specified layers and kernel sizes. Each block consists of upsampling, concatenation, dropout, convolutional layers, batch normalization, ReLU activation, and skip connections. The decoder takes an input of shape (256, 256, 1) and outputs the final reconstructed image of shape (256, 256, 1).

SampleMin-max normalization

```

def minmax_normalize(data):

    min_val = np.min(data)
    max_val = np.max(data)
    normalized_data = (data - min_val) / (max_val - min_val)
    return normalized_data

```

Sample Dice Coefficient Code

```
def dice_coefficient(y_true, y_pred, smooth=1e-7):
    intersection = tf.reduce_sum(y_true * y_pred)
    union = tf.reduce_sum(y_true) + tf.reduce_sum(y_pred)
    dice = (2.0 * intersection + smooth) / (union + smooth)
    return dice

def dice_loss(y_true, y_pred):
    loss = 1 - dice_coefficient(y_true, y_pred)
    return loss

model.compile(optimizer='adam', loss=dice_loss, metrics=[dice_coefficient])
```

The `dice_coefficient` function calculates the Dice coefficient, which is the ratio of the intersection to the union of the true and predicted masks. The `dice_loss` function calculates the Dice loss by subtracting the Dice coefficient from 1. Finally, the model is compiled using the Dice loss as the loss function and the Dice coefficient as a metric for evaluation.

6. CHAPTER SIX: EXPERIMENTAL RESULT AND DISCUSSION

In this chapter, the results of the experiments carried out on the datasets using the proposed architectures are presented. To provide a comprehensive analysis of the model's performance, both quantitative and qualitative evaluations are conducted. The quantitative evaluation involves the use of performance metrics such as accuracy, Loss, precision, sensitivity, dice similarity score, and specificity. On the other hand, the qualitative evaluation includes visual inspection of the model's output in terms of segmentation maps and a comparison with ground truth masks.

6.1 Quantitative Evaluation and Discussion

In this subsection, the predictive performance of the proposed and baseline methods are validated using the commonly used metrics, such as Dice Similarity Score, Mean IOU, accuracy, specificity, sensitivity, precision etc. Specifically, Figure 4-1, provides a comprehensive overview of the performance evaluation metrics for each of the proposed models.

6.1.1 Segmentation Result Analysis



Figure 6-1 Residual UNet and Deep UNet performance result

Based on the values in the result in the Figure 4-1, it can be seen that Residual UNet performed better Deep Unet than across of all matrices. These metrics suggest that both models performed well in segmenting brain tumors, but Residual UNet was to some extent more accurate in its predictions. The visual analysis of the output images generated by both models also supports this conclusion.

Accuracy: Specifically, the evaluation results revealed that Residual UNet models accurately identified 99.33% and 98.75% of positive instances in the ground truth as positive in the predicted segmentation for BraTS2020 and BraTS2019 respectively and Deep UNet 99.11% and 98.58% for BraTS2020 and BraTS2019 respectively. Based on the results obtained from our experiments, it can be concluded that both the DeepUNet and Residual UNet models exhibited high levels of accuracy in the two BRaTS datasets, with accuracy rates exceeding 99%.

Mean IoU: When we see the overlap between the predicted segmentation masks and the ground truth masks, the proposed Residual UNet scored 89.30% for BRaTs2020 and 84.71% for BRaTs2019. From the result the proposed approach has the ability to correctly generate tumorous Region of Interest (ROI) with high overlap. Whereas, the baseline Deep UNet 86.65% for BRaTs2020 and 83.29% for BRaTs2019 is less effective compared to proposed model.

Dice Similarity Score (DSC): Furthermore, in medical imaging, it is common for only a small portion of the image, known as the region of interest (ROI), to contain the target structure, while the remaining pixels are labeled as background. This situation can introduce bias during model training, as the model may disproportionately focus on the background pixels. To address this challenge, the Dice similarity score (DSC) or Dice coefficient is a more appropriate evaluation metric. The DSC quantifies the overlap between the ground truth and the segmented label, making it well-suited for assessing segmentation accuracy. In this study, both models achieved higher Dice similarity scores.

In our study the proposed model Residual model has achieved 91.85%, and 88.24% of DSC for BRaTs 2020 and BRaTs 2019 respectively, and baseline model Deep UNet has achieved 88.60% and 87.26% of DSC for BRaTs 2020 and BRaTs 2019 respectively. So, the proposed model has performed better than the baseline.

Precision: measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive, including both true positives and false positives. In simpler terms, precision answers the question: "Of all the instances predicted as positive, how many were actually positive?"

The Residual UNet and DeepUNet models have achieved a **precision** of 99.37% and 99.19% respectively for BRaTs 2020, and also those models achieved precision of 98.85% and 98.73% for BRaTs 2019 respectively. This indicates that both models accurately predicted positive instances, but the proposed model which is Residual UNet has performed better than the baseline which is Deep UNet.

6.1.2 Detection Result Analysis

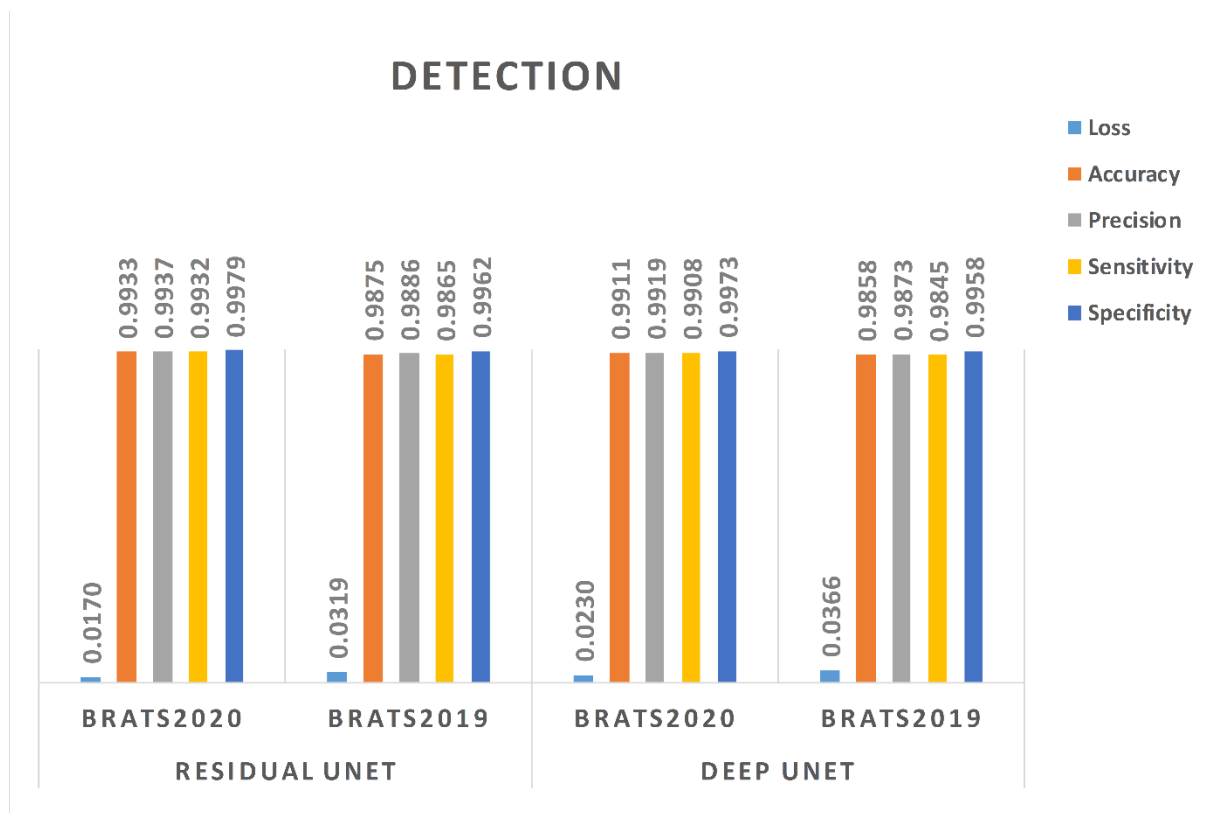


Figure 6-2 Residual UNet and DeepUNet detection performance result

Specificity: The Residual UNet and DeepUNet models achieved a **specificity** of 99.79% and 99.73% respectively in correctly identifying negative instances from the ground truth as negative in the predicted segmented label in the BRaTS2020 dataset, and also those models achieved specificity of 99.62% and 99.58% for BRaTs 2019 respectively. This indicates that both the

proposed model and the baseline model have a high level of proficiency in accurately predicting target structures in brain images.

Collectively, these findings underscore the efficacy of both models in accurately detecting positive and negative instances within the ground truth. Sensitivity and specificity metrics were employed to determine these results. Assessing the models using these metrics is crucial for determining whether all tumors have been successfully identified, which holds significant importance in diagnosis, surgery, treatment, and other applications.

However, when evaluating multiclass segmentation, it is not advisable to rely solely on accuracy, sensitivity, and specificity as the primary metrics. This is due to the issue of class imbalance. Medical images often contain a substantial number of true negative classes, leading to artificially high accuracy scores, even when an entire image is predicted as healthy. Therefore, alternative evaluation methods should be considered to appropriately assess the performance of the models in multiclass segmentation tasks.

Cross Entropy Loss:As can be seen from the result, the proposed Residual UNet has a cross-entropy loss of 0.0170 for BRaTs2020 and 0.0319 for BRaTs2019 dataset. Whereas, the baseline could achieve 0.0230 for BRaTs 2020 and 0.0366 for BRaTs 2019. Overall, the loss shows us that the proposed method has better performance in most of the metrics evaluated.

Sensitivity: measures the proportion of correctly detected objects out of all the ground truth objects. It is calculated as the ratio of true positives to the sum of true positives and false negatives (objects not detected by the model). According to this metrics the proposed model has achieved 99.32% and 98.65% for BRaTs 2020 and 2019 respectively, and the baseline which is Deep Unet also achieved 99.08% and 98.45% for BRaTs 2020 and 2019 respectively. So, the proposed model Residual model has achieved better than the baseline Deep Unet model.

6.1.3 Detection Rates

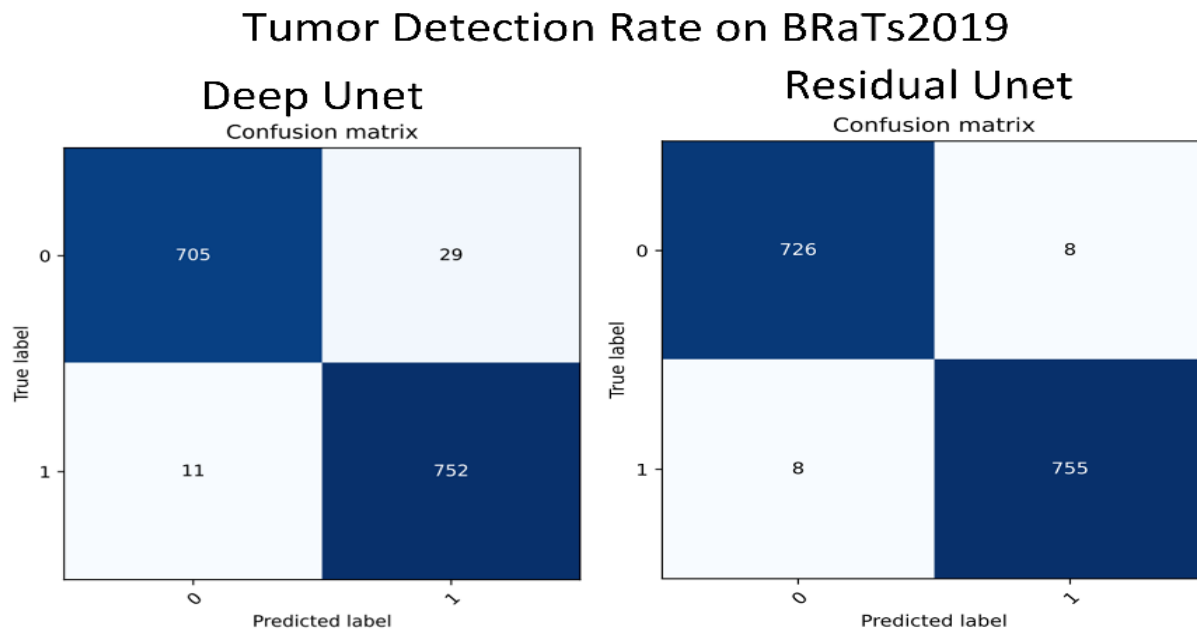


Figure 6-3. Tumor detection on BRaTs2019

Residual UNet model outperforms the baseline Deep UNet model in terms of tumor detection rate. The Residual UNet model achieves a high tumor detection rate with only 16 instances of confusion, while the Deep UNet model has 40 instances of confusion. Confusion refers to the misclassification of tumor and non-tumor regions. In this context, a lower number of confusions indicates that the Residual UNet model is better at accurately detecting tumors compared to the Deep UNet model.

Figure 6-3 shows the performance comparison between the two models on the BRaTs2019 dataset, which includes samples with and without tumors. The Residual UNet model demonstrates superior performance by achieving a higher tumor detection rate with fewer instances of confusion. This suggests that the proposed Residual UNet architecture is more effective in distinguishing between tumor and non-tumor regions compared to the baseline Deep UNet architecture.

Tumor Detection Rate on BRaTs2020

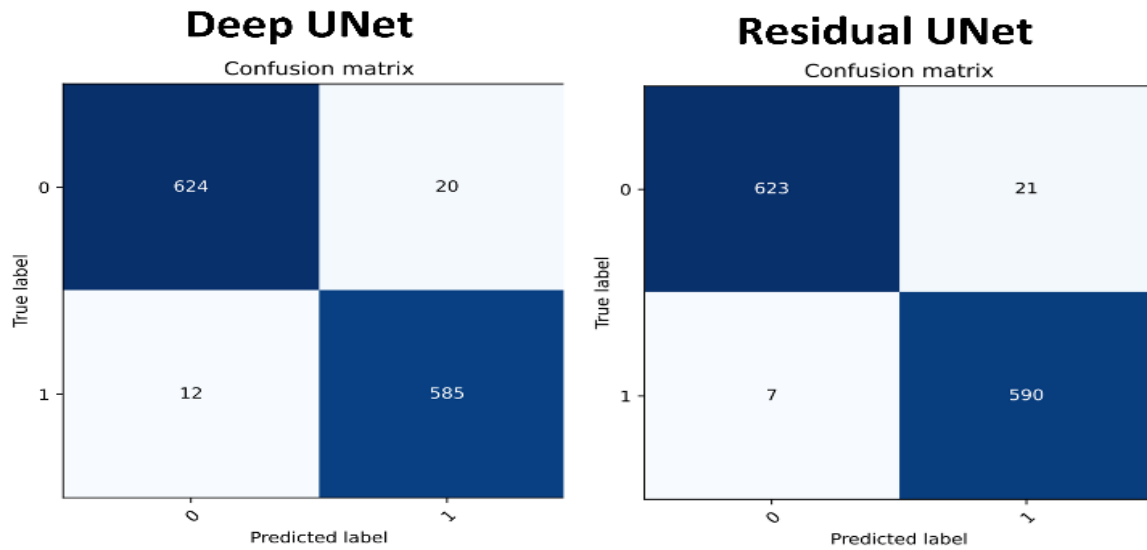


Figure 6-4. Tumor Detection Rate on BRaTs2020

Residual UNet model outperforms the baseline Deep UNet model in terms of tumor detection rate. The Residual UNet model achieves a high tumor detection rate with only 28 instances of confusion, while the Deep UNet model has 32 instances of confusion. Confusion refers to the misclassification of tumor and non-tumor regions. In this context, a lower number of confusions indicates that the Residual UNet model is better at accurately detecting tumors compared to the Deep UNet model.

Figure 4-3 shows the performance comparison between the two models on the BRaTs2020 dataset, which includes samples with and without tumors. The Residual UNet model demonstrates superior performance by achieving a higher tumor detection rate with fewer instances of confusion. This suggests that the proposed Residual UNet architecture is more effective in distinguishing between tumor and non-tumor regions compared to the baseline Deep UNet architecture.

6.2 Qualitative Experimental Results

The proposed model's performance was evaluated on two benchmark datasets, BRATS2019 and BRATS2020. To assess the performance of the proposed and baseline model, a visual analysis was carried out on the predicted segmented output images generated by the model. The results show that the proposed models are able to accurately detect and segment the majority of the structures in the MRI images, indicating the effectiveness of the proposed model.

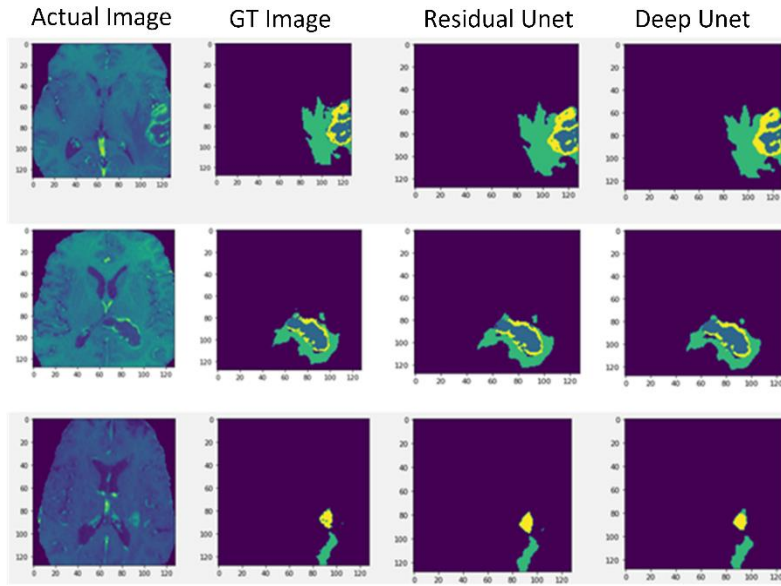


Figure 6-5 Predicted Results on BRATS2020 dataset

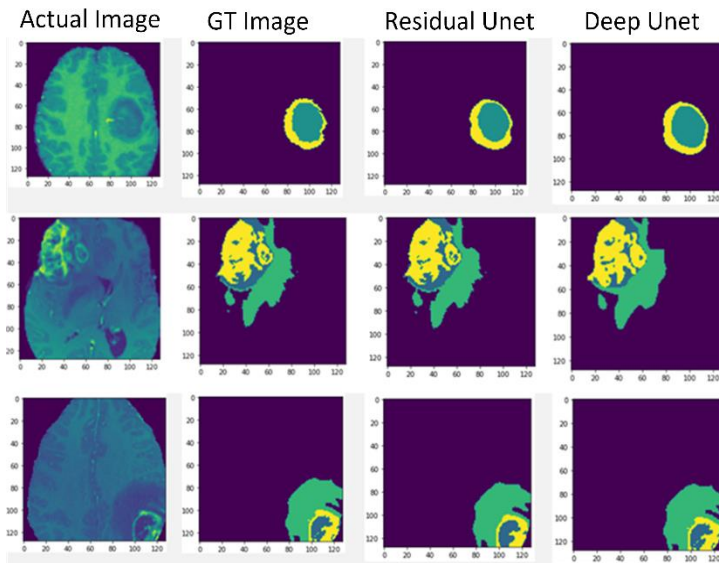


Figure 6-6 Predicted Results on BRATS2020 dataset

As shown in the above figures, both models are highly effective in identifying and detecting specific features of the tumor particularly the peritumoral area as well as the enhancing regions. The shape and location of the tumor areas are also given careful consideration by the models. The original image, which has a larger size, is reduced to 128×128 in order to make it computationally feasible for the model to process it. However, downsizing the image can cause information loss and certain details, such as some parts of a few tumors, may not be shown properly. To overcome this limitation, the model is applied to an image with a shape of 192×192 , which allows for more

details to be retained. A result obtained using Residual Unetmodel on a BRATS2020 sample with a size of 192×192 is being presented in the following figure.

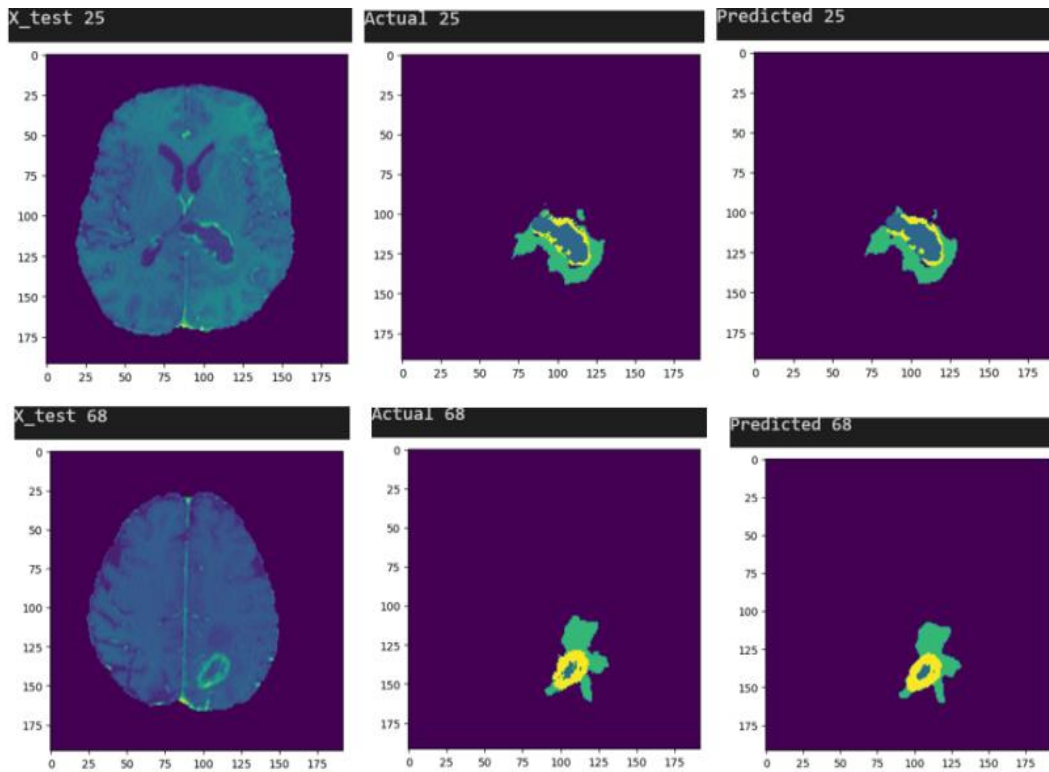


Figure 6-7 Residual UNet Results on an Image size of 192×192

While the model performs well on both image sizes, training on larger images like 192×192 can take longer due to the increased number of pixels that need to be processed. As a result, training on larger images may require more computational resources, such as a more powerful computer or longer training times, which can increase the cost and time needed to develop the model. However, the trade-off is that training on larger images can result in a detailed model that can better capture the complexity of the input images.

All of the experimental results reported above were obtained using the categorical cross-entropy loss function for training the models. Additionally, an experiment using the Dice loss function is also conducted. However, the performance of the model in predicting labels, except enhancing tumor and necrotic or non-enhancing tumor prediction, was found to be worse compared to models which were trained using the categorical cross-entropy loss function. The results showed that the model achieved Dice similarity score values of 0.900715, 0.895278, and 0.893918 for the training,

testing, and validation sets, respectively. These values indicate that the model is performing well in accurately segmenting the enhancing tumor class across the two sets of data.

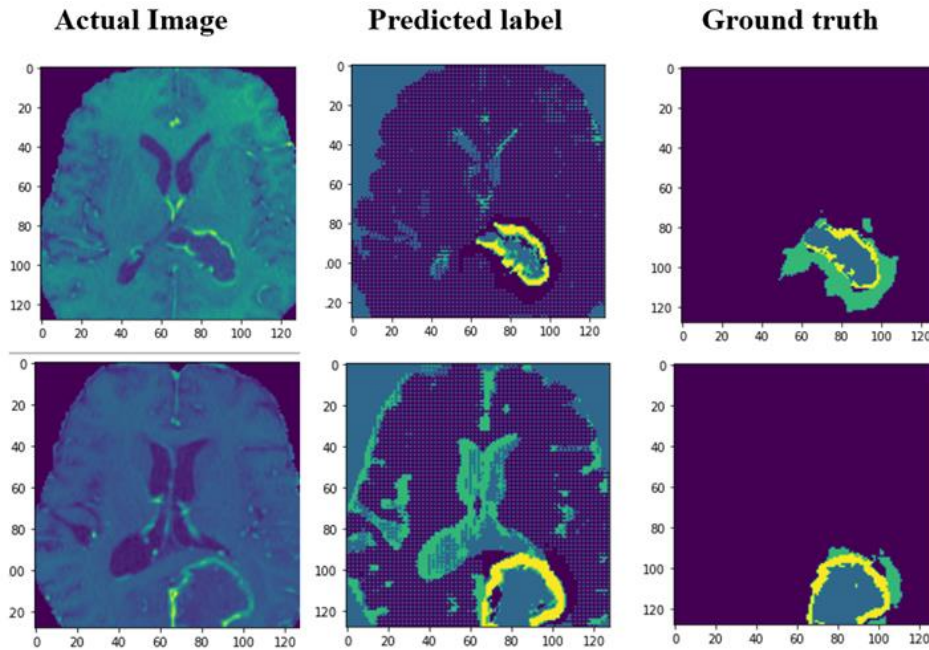


Figure 6-8 Residual UNet Performance using Dice loss

As shown above the model performs better for both enhancing and non-enhancing tumors. It is important to note that while these results are promising, further evaluation and analysis are required to fully assess the model's overall performance and generalizability to new and unseen data.

The Dice loss function is known to be particularly effective for image segmentation tasks and can help improve the accuracy of tumor segmentation. However, it may not be the best choice for all types of image segmentation tasks, as its performance on other label predictions may be suboptimal. Hence, it is crucial to meticulously evaluate and select the appropriate loss function, taking into account the unique requirements and attributes of the task.

6.3 Training, Validation and Loss Matrices Curves

The performance of the proposed models is evaluated using the metrics that are discussed in chapter 3. The following graph represents both the optimization (i.e. loss) and performance learning curves of the model on the training and validation sets using the both datasets. The training learning curves are obtained by calculating metric results on the same data that the model was

trained on, while the validation learning curves are obtained by calculating metrics based on the performance of the model's predictions on new data.

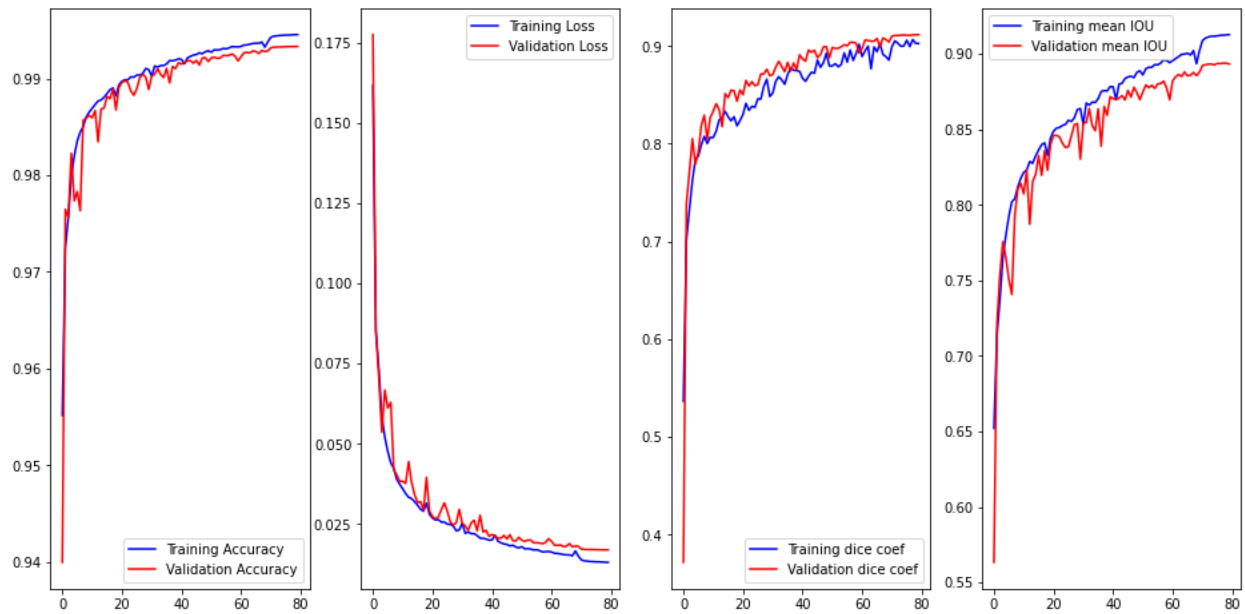


Figure 6-9 Residual UNet Learning Curve on BRATS2020 Dataset

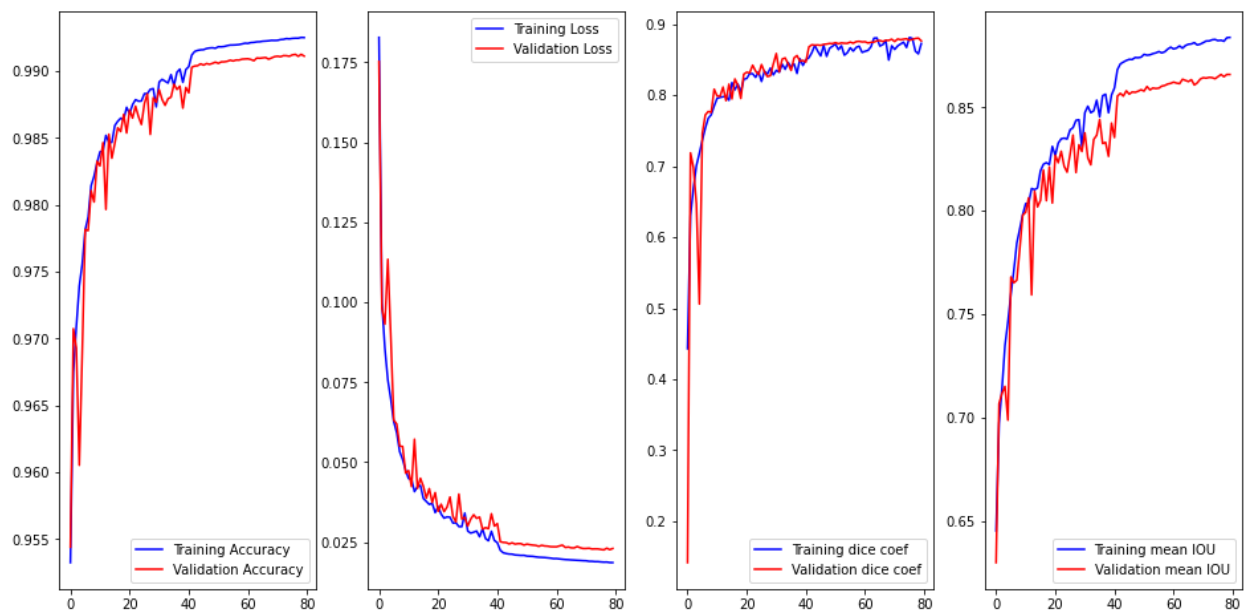


Figure 6-10 DeepUNet Learning Curve on BRATS2020 Dataset

The essential aim of both validation and training learning curve is to Achieve a decreasing loss as the number of epochs progresses. During training, the loss function holds paramount importance as it directly reflects the network's learning process, superseding other metrics. As shown in the

above curves, the validation and training loss drops down in each epoch which indicates the model is well trained. In addition, the stability and dynamics of both learning curves are used to understand the behavior of the model. When a model performs well on the training data but poorly on the validation data, it is said to be overfit. This can happen due to the model's architectural complexity or long training period. Over fitting can also occur when the model learns noise or irrelevant information in the dataset by training for an excessively long time or being overly complex. Under these circumstances, the validity loss decreases up until a certain point and then begins to increase once more. To address this issue, two regularization methods, namely early stopping and dropout, are commonly used. The dropout function periodically ignores or “drops out” a certain number of layer outputs to reduce the complexity of the model. The early stopping strategy seeks to stop the model's training at a specific point before it starts memorizing noise and random changes in the input. The learning curves of both models on BRATS2019 ⁴are illustrated below.

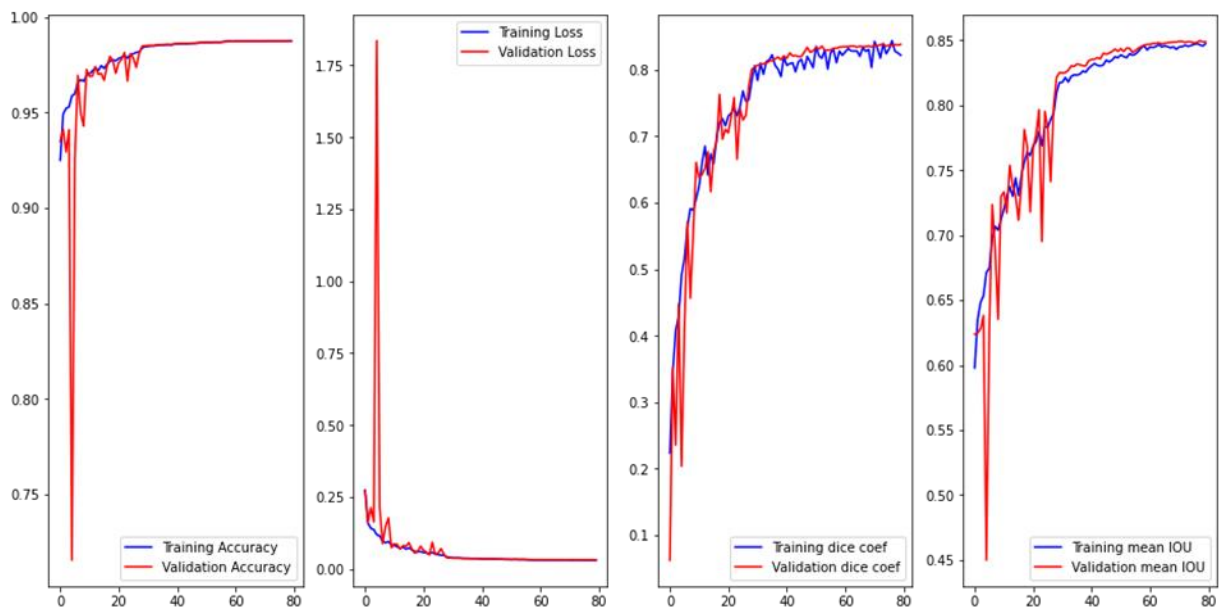


Figure 6-11 Residual Unet Learning Curve on BRATS2019 Dataset

⁴<https://www.kaggle.com/datasets/aryashah2k/brain-tumor-segmentation-brats-2019>

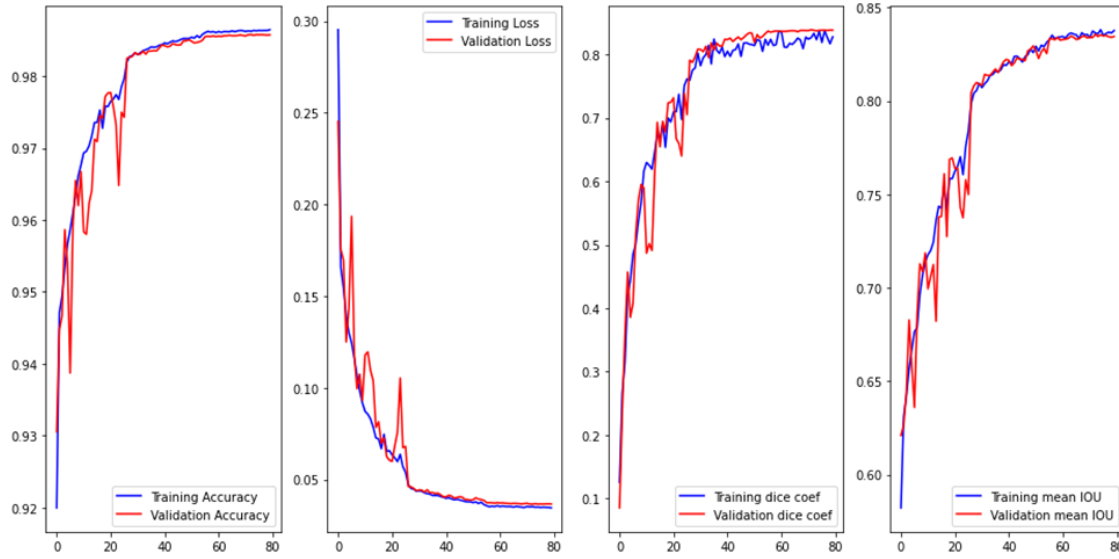


Figure 6-12 Deep Unet Learning Curve on BRATS2019 Dataset

6.4 Comparison with State-of-the-art Works

The segmentation performance of the proposed method is compared with recent state-of-the-art methods on the two BraTS benchmaks: BraTS2019 and BraTS2020. Specifically, Nested-UNet (Nguyen, H. T., Le, T. T., Nguyen, T. V., & Nguyen, N. T, 2020)), 3D-UNet (3D U-Net Segmentation, 2019), Ensembl-UNets (Ensemble of Updated-Unets), MvSSS (Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao and Xirong Li, Member, IEEE, 2022)], MECU-Net (Xinchao Cheng,Zongkang Jiang, 2019). Ensembl-2DCN (Shah, A. A., Malik, H. A. M., Muhammad, A., Alourani, A., & Butt, Z. A., 2023)], Cascad-UNet (Hongying Liu, Xiongjie Shen, Fanhua Shang, Fei Wang, 2019), MM-UNet (Liang Zhao,Jiajun Ma,Hong Yuan,Chaoran Jia, 2022), are used for fair comparison on the three BraTS datasets.

Additionally, understanding the location and extent of the enhancing tumor can help in surgical planning to remove as much of the tumor as possible while minimizing damage to healthy brain tissue. In general, evaluating the performance of the proposed methods on whole tumor, tumor core, and enhancing tumor allows the medical professional to better understand the extent and location of the tumor, which can aid in treatment planning and monitoring.

Table 4-5 presents the results of the proposed methods as well as other state-of-the-art works on the test data of BRaTS2020. Based on the segmentation results in Table 4-5, the proposed

DeepUNet and Residual UNet have achieved remarkable performance improvements consistently across the three challenging BraTS benchmark datasets.

Table 6-1 State of arts comparison

Metrics Models	Tumor	Dataset
MvSSS (Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao and Xirong Li, Member, IEEE, 2022)	0.883	BRaTS2019
MECU-Net (Xinchao Cheng,Zongkang Jiang, 2019)	0.890	
Ensembl-2DCNN (Shah, A. A., Malik, H. A. M., Muhammad, A., Alourani, A., ... & Alotaibi, 2022)	0.8865	
Cascad-UNet (Hongying Liu, Xiongjie Shen, Fanhua Shang, Fei Wang, 2019)	0.88796	
Nested Unet (Nguyen, H. T., Le, T. T., Nguyen, T. V., & Nguyen, N. T, 2020)	0.8999	
3D Unet (Henry, T., Carré, A., Lerousseau, M., Estienne, T., Robert, C., Paragios, N., & Deutsch, E, 2021)	0.91148	
Ensemble Unets (Ensemble of Updated-Unets)	0.9103	
Proposed Residual Unet	0.9875	
Nested Unet (Baid, U., Ghodasara, S., Mohan, S., Bilello, M., Calabrese, E., Colak, E., ... & Bakas, S., 2021)	0.8999	BRaTS2020
3D Unet (Ahmad, P., Qamar, S., Shen, L., ... & Wang, 2021)	0.91148	
Ensemble Unets (Zhang, Y., Zhong, P., Jie, D., Wu, J., Zeng, S., Chu, J., ... & Tang, X., 2021)	0.9103	

Proposed Residual Unet	0.9933	
-------------------------------	--------	--

BraTS2019. As can be seen from the table, the segmentation accuracy obtained by the proposed Residual UNet on the BraTS2019 dataset out performed all existing methods by a significant margin. For instance, Residual UNet outperforms 3D-UNet (Henry, T., Carré, A., Lerousseau, M., Estienne, T., Robert, C., Paragios, N., & Deutsch, E, 2021)] by +7.6% (91.1% vs. 98.75%) accuracy on whole tumor segmentation task.

BraTS2020. Similarly, the proposed methods achieved promising segmentation results on BraTS2020 compared to the existing methods. Residual UNet attains a significant performance improvement across the whole, enhanced, and core segmentation matrices. Specifically, the proposed Residual UNet outperforms the best performing Ensemble-UNets (Zhang, Y., Zhong, P., Jie, D., Wu, J., Zeng, S., Chu, J., ... & Tang, X., 2021) by +8.3 % larger margin on BraTS2020 dataset.

Discussion. In compare to the above-mentioned state of art works, it is more clear that our model outperformed the state-of-the-art models on both datasets. The model's impressive accuracy score indicates its efficacy for automated brain tumor segmentation, making it a promising tool in this domain. The model's capability to accurately identify small, subtle features in the images is particularly noteworthy, as this is a challenging task that can be difficult for human experts to perform consistently. Furthermore, the high precision and recall rates indicate that the model can effectively identify both positive and negative instances, which is important for ensuring accurate diagnosis and treatment planning.

6.5 Contribution of the Study

The proposed architecture, the Residual UNet combines the U-Net structure with residual connections. The residual connections allow for the direct flow of information from the encoder to the decoder, enabling the model to access both low-level and high-level features. This facilitates the extraction of abstract and global contextual information, as the model can leverage the features learned at different spatial scales.

The skip connections in the Residual U-Net enable the reuse of features learned at different scales. This feature reuse mechanism helps in better localization and segmentation of objects in the image by combining both global and local information. They allow for direct propagation of gradients, reducing the impact of vanishing gradients and enabling faster convergence during training.

In the context of multiclass segmentation, it's common to use deeper architectures with more layers to capture more complex features and enhance the model's capacity to learn intricate patterns. However, in our model we found that a four-layer residual unit performs well for segmentation and detection task, it indicates that the complexity of the problem can be adequately captured with a simpler model.

6.6 Research question Discussions

According to the research questions asked in chapter one the following discussions have made.

Answer for Q1: The proposed architecture Residual Unet enhances the efficiency of brain tumor detection and segmentation compared to traditional methods by utilizing skip connections and residual connections. These connections help retain important information from earlier layers, addressing vanishing gradients and enabling precise tumor boundary localization. The model also benefits from enhanced global information processing through the encoder and decoder pathways, capturing both local and global context for improved performance. Overall, the Residual UNet model improves efficiency through these techniques, addressing challenges and enhancing segmentation and detection performance.

Answer for Q2: Employing different datasets when training and evaluating deep neural network models for brain tumor detection and segmentation offers several benefits. Tumors can exhibit significant variations in terms of size, shape, location, and appearance. By training the model on different datasets, it becomes more robust and capable of handling various tumor characteristics, leading to improved generalization and performance on unseen data. Additionally, employing different datasets enables the validation and comparison of model performance against existing methods and benchmarks. By evaluating on multiple datasets, we can assess the generalizability and effectiveness of their proposed model compared to other state-of-the-art approaches. This

approach also enhances the model's robustness, allowing it to perform well across various tumor characteristics, patient populations, and imaging protocols.

Answer for Q3:When comparing the performance of a The proposed Residual unet model for brain tumor detection and segmentation to other state-of-the-art works, several performance improvements have been achieved. These improvements observed through various evaluation metrics, demonstrating the model's superior capabilities.

7. CHAPTERSEVEN: CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this thesis, a deep learning-based brain tumor segmentation technique are proposed that can accurately identify and segment different types of brain tumors from MRI scans. These models address the limitations of the UNet model, which has a small receptive field of convolutional layers, and are designed to overcome the challenges of brain tumor detection and segmentation.

The models underwent training using two publicly accessible brain tumor detection and segmentation challenge datasets, namely BRATS 2019 and BRATS 2020, and were preprocessed using normalization and patching techniques. A min-max scaling is used to ensure that the input images have consistent intensity values and contrast, which is necessary for the model to learn meaningful features from the data. Patching, on the other hand, involves dividing the input image into smaller patches to reduce the memory requirements of the model and enable it to learn from larger datasets.

The proposed model improves upon the UNet architecture by embedding a residual skip connection module that allows for the extraction of both local and global contexts, and increasing the depth of the encoder network to extract a larger number of abstract features, leading to improved segmentation accuracy.

The proposed method, Residual Unet is evaluated using common evaluation metrics such as Dice similarity coefficient (DSC), precision, sensitivity, and specificity, and found that it outperforms several other state-of-the-arts and the baseline model in brain tumor detection and segmentation methods on the same datasets.

This study makes a valuable contribution to the advancement of more precise and dependable methods for brain tumor detection and segmentation. These methods have the potential to aid radiologists and clinicians in diagnosing brain tumors, ultimately leading to improved patient outcomes and quality of life. Our research showcases the effectiveness of deep learning-based

approaches in accurately detecting and segmenting brain tumors and emphasizes the significance of further advancements in this field to enhance patient care.

7.2 Future Work

Due to memory constraints, the models were only able to train on small amount data samples. In the future works, a data augmentation technique will be explored to increase the data size and enhance the model's generalization capability. In addition, it is aimed to incorporate 3D information into our models, as the current approach only utilizes 2D planes of MRI images. Incorporating 3D information can potentially lead to a more precise detection and segmentation of brain tumors. There is also plan to investigate the use of transfer learning and other advanced deep learning techniques to further improve the accuracy and efficiency of our models. These efforts will ultimately contribute to the development of more effective and robust brain tumor detection and segmentation techniques that can better support clinical decision-making and patient care.

Reference

- Al-Qazzaz, S. (2020). Deep learning-based brain tumour image segmentation and its extension to stroke lesion segmentation. *Online Research @ Cardiff* .
- Amjad Rehman Khan, Siraj Khan, Majid Harouni, Rashid Abbasi, Sajid Iqbal, Zahid Mehmood. (2021). Brain tumor segmentation using K-means clustering and deep learning with synthetic data augmentation for classification. *Wiley Analytical Science* , 1389-1399.
- Anaraki, A. K., Ayati, M., & Kazemi, F. (2019). Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *biocybernetics and biomedical engineering* , 63-74.
- Arkapravo Chattopadhyay, M. M. (2022). MRI-based brain tumour image detection using CNN based deep learning method. *ScienceDirect* .
- Ayesha Younis, Li Qiang Charles Okanda Nyatega, Mohammed Jajere Adamu and Halima Bello Kawuwa. (2022). Brain Tumor Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches. *Applied sciences* .
- Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding Batch Normalization. *NeurIPS proceedings* , 31.
- Caliva, F., Iriondo, C., Martinez, A. M., Majumdar, S., & Pedoia, V. (2019). Distance map loss penalty term for semantic segmentation. *arXiv preprint arXiv* .
- D C Febrianto, I Soesanti, H A Nugroho. (2020). Convolutional Neural Network for Brain Tumor Detection. *2nd International Conference on Engineering and Applied Sciences (2nd InCEAS)* (pp. doi:10.1088/1757-899X/771/1/012031). Yogyakarta: IOP Publishing.
- Dr. J. Jegathesh Amalraj, S. Banumathi, J. Jereena John. (2019). IOT Sensors And Applications: A Survey. *08* (08).
- Grover, V. P., Tognarelli, J. M., Crossey, M. M., Cox, I. J., Taylor-Robinson, S. D., & McPhail, M. J. (2015). Magnetic resonance imaging: principles and techniques: lessons for clinicians. *Journal of clinical and experimental hepatology* , 246-255.

Guo, Y., Cao, X., Liu, B., & Gao, M. . (2020). Cloud detection for satellite imagery using attention-based U-Net convolutional neural network. *symmetry* , 1056.

Hamza Rafiq Almadhoun and Samy S. Abu Naser. (2022). Detection of Brain Tumor Using Deep Learning. *International Journal of Academic Engineering Research (IJAER)* , :29-47.

Hedaoo, A. H. (2021). Deep Learning and its Application: A Review. *International Journal of Research Publication and Reviews* , 340-352.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. (2015). Deep Residual Learning for Image Recognition. *Microsoft Research* .

Lingling Fang, Xin Wang. (2022). Brain tumor segmentation based on the dual-path network of multi-modal MRI images. *Elsevier* .

Masood M, Nazir T, Nawaz M, Mehmood A, Rashid J, Kwon HY, Mahmood T, Hussain A. (2021). A Novel Deep Learning Method for Recognition and Classification of Brain Tumors from MRI Images. *Diagnostics* . *Diagnostics (Basel)* , 744.

Mohammad Attique Kahan,Imran Ashraf,Majed Alhaisoni,ertas Damaševičius,Rafal Scherer,Amjad Rehman,Syed Ahmad Chan Bukhari. (2020). Multimodal Brain Tumor Classification Using Deep Learning and Robust Feature Selection: A Machine Learning Application for Radiologists. *MDPI* .

Olaf Ronneberger, Philipp Fischer, Thomas Brox. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arxiv logo* .

P Gokila Brindha,M Kavinraj ,P Manivasakam,and P Prasanth. (2020). Brain tumor detection from MRI images using deep learning. *IOP Conf. Series: Materials Science and Engineering* . IOP Publishing.

Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., & Mehmood, Z. (2020). A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks. *Journal of medical systems* , 1-16.

Ranjbarzadeh, R., Bagherian Kasgari, A., Jafarzadeh Ghouschi, S. et al. (2021). Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images. *Scientific Reports* .

Roohollah Aslanzadeh, Kazem Qazanfari, Mohammad Rahmati. (2017). An Efficient Evolutionary Based Method For Image Segmentation. *arxiv* .

Sakshi Indolia, Anil Kumar Goswami, S. P. Mishra, Pooja Asopa. (2018). Conceptual Understanding of Convolutional Neural Network- A. *International Conference on Computational Intelligence and Data Science (ICCIDS 2018)* (pp. 679–688). Delhi: ScienceDirect.

Salwa Khalid Abdulateef and Mohanad Dawood Salman. (2021). A Comprehensive Review of Image Segmentation . *Iraqi Journal to Electrical and Electronic Engineering* , 166-175.

Sathananthavathi V., Indumathi G. (2021). Encoder Enhanced Atrous (EEA) UNet architecture for Retinal Blood vessel segmentation. *ScienceDirect* , 84-95.

Schmidhuber. (2015). Deep learning in neural networks: An overview. *Science Direct* , 85-117.

The Johns Hopkins University, The Johns Hopkins Hospital, and Johns Hopkins Health System. (2022). Brain Tumors and Brain Cancer. *JOHNSHOPKINS medicine* .

Tonmoy Hossain; Fairuz Shadmani Shishir; Mohsena Ashraf; MD Abdullah Al Nasim; Faisal Muham. (2019). Brain Tumor Detection Using Convolutional Neural Network. *IEEE* , 1-6, doi.

Triplanar Ensemble of 3D-to-2D CNNs with Label-Uncertainty for Brain Tumor Segmentation. (2020). *International MICCAI Brainlesion Workshop* (pp. 379–387). University of Pennsylvania, Philadelphia, PA, USA: Springer, Cham.

Vivekraj A., Sumathi s. (2023). Resnet-Unet-FSOA based cranial nerve segmentation and medial axis extraction using MRI images. *Taylor Francis Online* .

Wang, H., Mingye, X., Chen, X. et al. (2022). Featured article: Radiomics analysis of contrast-enhanced computed tomography in predicting the International Neuroblastoma Pathology Classification in neuroblastoma. *Insights into Imaging* .

Wang, W., Chen, C., Ding, M., Yu, H., Zha, S., & Li, J. (2021). Multimodal brain tumor segmentation using transformer. In *Medical Image Computing and Computer Assisted*

Intervention. *24th International Conference* (pp. 109-119). Strasbourg: Springer International Publishing.

Wang, W., Xie, E., Li, X., Fan, D. P., Song, K., Liang, D., ... & Shao, L. . (2022). Improved baselines with pyramid vision transformer. *Computational Visual Media*, , 415-424.

Xu, Dan and Zhou, Xidong and Niu, Xuefen and Wang, Junwei. (2020). Automatic segmentation of low-grade glioma in MRI image based on UNet++ model. *IOP Publishing* (p. 012135). Journal of physics.

Xue, Z., Jiang, M., Liu, H., Zeng, S., & Hao, J. (2020). Low dose soft X-ray-controlled deep-tissue long-lasting NO release of persistent luminescence nanoplatfrom for gas-sensitized anticancer therapy. *Biomaterials* , 263.