ADVANCED GRAPH CONVOLUTION NETWORKS
FOR
MRI-BASED BRAIN TUMOR SEGMENTATION

A PROJECT REPORT (CS6811)

Submitted by

PARI PS (2021103029)

KESAVARAJAMANIKANDAN K (2021103733)

PAL DHINAKARAN P (2021103736)

KAVISHKUMAR K (2021103321)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING, GUINDY CAMPUS

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**ADVANCED GRAPH CONVOLUTION NETWORKS FOR MRI-BASED BRAIN TUMOR SEGMENTATION”** is the bonafide work of “**PARI PS (2021103029), KESAVARAJAMANIKANDAN K (2021103733), PAL DHINAKARAN P (2021103736) and KAVISHKUMAR K (2021103321)**” who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these or any other candidates.

Place : Chennai

Date : 22/04/2025

SIGNATURE

Dr. V. Mary Anita Rajam

HEAD OF THE DEPARTMENT

Computer Science and Engineering,

Anna University, Chennai - 600 025.

SIGNATURE

Dr.B.L.VELAMMAL

SUPERVISOR

Associate Professor

Computer Science and Engineering,

Anna University, Chennai – 600 025.

ACKNOWLEDGEMENT

We express our deep gratitude to our guide, **Dr. B.L. Velammal**, Associate Professor, Department of Computer Science and Engineering, for guiding us through every phase of the project. We appreciate her thoroughness, tolerance and ability to share her knowledge with us. We would also like to thank his /her for her kind support and for providing necessary facilities to carry out the work.

We are extremely grateful to **Dr. V. Mary Anita Rajam**, Professor & Head of the Department, Department of Computer Science and Engineering, Anna University, Chennai – 25, for extending the facilities of the Department towards our project and for her unstinting support.

Our express our thanks to the panel of reviewers **Dr.S.Bose**, Professor, Department of Computer Science and Engineering, **Dr.S.Sudha**, Professor, Department of Computer Science and Engineering, for their valuable suggestions and critical reviews throughout the course of our project.

We express our thanks to all other teaching and non-teaching staff who helped us in one way or other for the successful completion of the project. We would also like to thank our parents, family and friends for their indirect contribution in the successful completion of this project.

PARI PS   KESAVARAJAMANIKANDAN K   PAL DHINAKARAN P   KAVISHKUMAR K

ABSTRACT – ENGLISH

Brain tumor detection and segmentation play a critical role in the diagnosis and treatment planning of gliomas. Traditional convolutional neural networks (CNNs) often struggle to effectively capture the complex spatial and inter-modal relationships inherent in multi-modal MRI data. To address this limitation, we propose a novel framework named M2GCNet (Multi-modal Graph Convolutional Network) for accurate brain tumor detection and segmentation. M2GCNet integrates both spatial and channel-wise graph convolution modules to exploit local and global contextual features while preserving structural and semantic relationships across different MRI modalities (T1, T1c, T2, and FLAIR). The architecture consists of preprocessing, feature extraction, spatial and channel graph construction, multi-modal correlation learning, and a fusion-based segmentation decoder. A key component of our model is the Multi-modal Correlation Loss, which ensures coherent feature alignment between modalities, enhancing the network's discriminative power. We evaluate our model on the BraTS 2018 and BraTS 2019 datasets, achieving competitive results in tumor core, whole tumor, and enhancing tumor segmentation tasks. The proposed approach demonstrates superior performance in leveraging multi-modal data through graph-based modeling, making it a promising solution for brain tumor detection and clinical decision support.

ABSTRACT – TAMIL

மூளை நுரை உருவாகும் நோயின் கண்டறிதல் மற்றும் பிரித்தல், சரியான நோயறிதல் மற்றும் சிகிச்சை திட்டமிடலில் முக்கிய பங்காற்றுகிறது. பாரம்பரிய கன்வல்யூஷனல் நரம்பியல் வலையமைப்புகள் (CNNs), பன்முகப்படக்காட்சிகள் கொண்ட MRI தரவுகளில் உள்ள சிக்கலான இட மற்றும் மாதிரித்தொகுப்புகளின் உறவுகளை முழுமையாகப் பயன்படுத்த முடியாமல் தவிக்கின்றன. இந்த சிக்கலை சமாளிக்க, நாங்கள் M2GCNet (பன்முகப்படக்காட்சிகளுக்கான கிராஃப் கன்வல்யூஷனல் நெட்வொர்க்) என்ற புதிய மாடலை முன்வைக்கிறோம். இது, இடைவழி மற்றும் சேனல் அடிப்படையிலான கிராஃப் கன்வல்யூஷன்களை இணைத்து, MRI மாதிரிகளில் உள்ள உள்ளூர் மற்றும் உலகளாவிய தன்மைகளை கையாள்கிறது.

இந்த கட்டமைப்பில் முன்பக்க செயலாக்கம், அம்சவிளக்கவியல், இடவழி மற்றும் சேனல் கிராஃப் கட்டமைப்பு, பன்முகப்படக்காட்சி தொடர்பு இழப்பு, மற்றும் ஒரு இணைப்பு அடிப்படையிலான பிரிப்பு டிகோடர் ஆகியவை அடங்கும். முக்கிய அம்சமாக, பன்முகப்படக்காட்சி தொடர்பு இழப்பு (Multi-modal Correlation Loss) மூலம் ஒவ்வொரு மாதிரிக்கும் இடையே உள்ள அம்சங்கள் ஒத்திசைவாக இருக்கின்றன, இது நெட்வொர்கின் வேறுபாடுகளைக் கண்டறியும் திறனை மேம்படுத்துகிறது.

BraTS 2018 மற்றும் BraTS 2019 தரவுத்தொகுப்புகளில் இந்த மாடலை சோதித்தபோது, முழுமையான நுரை, நுரை மையம், மற்றும் மேம்பட்ட நுரை பிரிப்புகளில் மிகச்சிறந்த செயல்திறனை வெளிப்படுத்தியது. பன்முகப்படக்காட்சி தரவுகளை கிராஃப் அடிப்படையில் சிறப்பாக செயல்படுத்தும் இந்த முறை, மருத்துவத் துறையில் ஒரு வலிமையான தீர்வாக திகழ்கிறது.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT – ENGLISH	**iii**
	ABSTRACT – TAMIL	**iv**
	LIST OF TABLES	**vii**
	LIST OF FIGURES	**viii**
	LIST OF ABBREVIATIONS	**xi**
1.	**INTRODUCTION**	1
	1.1 INTRODUCTION	1
	1.2 OBJECTIVES	2
	1.3 PROBLEM STATEMENT	3
	1.4 CHALLENGES & APPLICATIONS	4
	1.5 METHODOLOGY OVERVIEW	6
	1.6 ORGANISATION OF THESIS	8
2.	**LITERARY REVIEW**	
	2.1 DEEP LEARNING TECHNIQUES IN BRAIN TUMOR SEGMENTATION	9
	2.2 GRAPH CONVOLUTIONAL NETWORKS IN MEDICAL IMAGING	9
	2.3 MULTI-MODAL MRI DATA FUSION	10
	2.4 ADVANCED SEGMENTATION NETWORKS	10
	2.5 MULTI-MODAL CORRELATION	11

LEARNING

2.6 LIMITATIONS IN EXISTING METHODS 11

3. SYSTEM DESIGN AND IMPLEMENTATION

3.1 SYSTEM ARCHITECTURE 13

3.2 PROPOSED SYSTEM 14

3.3 MODULE 1 – DATA PREPROCESSING 14

3.4 MODULE 2 – FEATURE EXTRACTION 17

3.5 MODULE 3 – ENCODER 19

3.6 MODULE 4 – GRAPH CONVOLUTION MODULES 21

3.7 MODULE 5 – DECODER 22

3.8 MODULE 6 – SEGMENTATION 24

4. RESULTS AND DISCUSSION

4.1 DATASET DESCRIPTION 25

4.2 RESULTS 26

4.3 TEST CASES 28

4.4 PERFORMANCE EVALUATION 30

5. CONCLUSION

5.1 CONCLUSION 32

5.2 FUTURE WORKS 32

6. REFERENCES 34

LIST OF TABLES

Table No.	Description	Pg. No
4.1	Dataset description	1
4.2	Test cases	3

LIST OF FIGURES

Figure No.	Description	Pg. No
3.1	Overall Architecture Diagram	13
3.3	Module 1 - Data preprocessing	15
3.4	Module 2 - Feature Extraction	17
3.5	Module 3 - Encoder	19
3.6	Module 4 - Graph Convolution Module	21
3.7	Module 5 - Decoder	22
3.8	Module 6 - Segmentation	24

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
AE	Auto Encoders
M2GCNET	Multi Model Graph Convolutional Network
MRI	Magnetic Resonance Imaging
CGCM	Channel Graph Convolution Module
SGCM	Spatial Graph Convolution Module
WT	Whole Tumor
TC	Tumor Core
FLAIR	Fluid-Attenuated Inversion Recovery
T1	T1-weighted MRI
T1c	T1-weighted MRI with contrast enhancement
T2	T2-weighted MRI

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Brain tumor detection and segmentation are critical tasks in medical image analysis, enabling accurate diagnosis and treatment planning. Magnetic Resonance Imaging (MRI) provides detailed multi-modal information (T1, T1c, T2, and FLAIR), but efficiently extracting and utilizing this data for precise tumor segmentation remains a challenge due to modality variation and complex spatial features.

This project proposes a novel deep learning architecture called M2GCNet (Multi-modal Graph Convolutional Network), which effectively integrates Convolutional Neural Networks (CNNs) with Graph Convolutional Networks (GCNs) to enhance segmentation performance. The model is designed to capture both spatial and channel-wise relationships between features extracted from different MRI modalities.

The overall system comprises the following modules:

- Preprocessing Module: Normalizes and augments the multi-modal MRI inputs.
- Feature Extraction Module: Extracts rich features using CNNs.
- Encoder Module: Encodes spatial hierarchies through downsampling.
- Graph Convolution Module: Constructs spatial and channel graphs to model complex inter-modal dependencies using SGCM and CGCM.
- Decoder and Segmentation Module: Reconstructs high-resolution segmentation maps with accurate tumor boundaries.

- Multi-modal Correlation Loss: Ensures feature alignment and improves segmentation consistency across modalities.

The proposed framework was evaluated using BraTS 2018 and BraTS 2019 datasets and demonstrated superior performance in segmenting the Whole Tumor (WT), Tumor Core (TC), and Enhancing Tumor (ET) regions. The integration of graph-based learning and multi-modal correlation significantly improved both accuracy and robustness, making the system suitable for clinical deployment.

1.2 OBJECTIVES

1. **To develop an automated system for accurate brain tumor detection and segmentation**
   Utilizing deep learning techniques to minimize manual intervention in medical diagnosis.
2. **To design a Multi-modal Graph Convolutional Network (M2GCNet)**
   Integrating both spatial and channel-wise graph convolutions to capture complex relationships between different MRI modalities.
3. **To preprocess multi-modal MRI data effectively**
   Including normalization, resizing, skull stripping, and augmentation for improved training efficiency and model generalization.
4. **To extract deep spatial features using convolutional layers**
   Employing an encoder architecture to learn hierarchical feature representations from T1, T1c, T2, and FLAIR modalities.
5. **To construct spatial and channel graphs from extracted feature maps**
   Enabling the model to learn both intra- and inter-modality dependencies that traditional CNNs might overlook.
6. **To implement a decoder with skip connections for high-resolution segmentation**

Reconstructing accurate tumor boundaries through upsampling and feature fusion.

7. **To introduce a Multi-modal Correlation Loss function** Aligning feature spaces across different modalities and enhancing segmentation performance.
8. **To evaluate the model on benchmark datasets like BraTS 2018 and BraTS 2019** Using metrics such as Dice Coefficient, Sensitivity, and ROC-AUC to assess effectiveness.
9. **To provide a user-friendly interface for tumor visualization** Allowing clinicians to upload MRI scans and view segmentation outputs interactively.

1.3 PROBLEM STATEMENT

Brain tumor segmentation is a critical task in medical image analysis, aiding in diagnosis, treatment planning, and patient monitoring. The Brain Tumor Segmentation (BraTS) challenge provides multi-modal MRI datasets that include structural MRI sequences such as T1, T1c, T2, and FLAIR. These datasets contain annotated tumor regions, including enhancing tumor (ET), tumor core (TC), and whole tumor (WT), which pose significant challenges due to variations in shape, size, intensity, and location. Traditional methods, such as manual segmentation by radiologists, are time-consuming and prone to inter-observer variability, necessitating the development of automated and efficient segmentation techniques.

Graph-based deep learning models, particularly Graph Convolutional Networks (GCNs), have shown promise in capturing complex spatial relationships within medical images. Multi-modal Graph Convolutional Networks (M2GCNet) leverage inter-modal dependencies to enhance segmentation accuracy by integrating spatial and contextual information across different MRI modalities. However, challenges such as class imbalance, small dataset sizes, and computational complexity remain significant hurdles. Addressing these challenges requires robust feature extraction, efficient graph construction, and optimization strategies to improve segmentation performance while maintaining low computational overhead.

This work focuses on leveraging M2GCNet for brain tumor segmentation using the BraTS 2018 and BraTS 2019 datasets. The aim is to refine model performance through optimized graph-based representations and improved training strategies while ensuring generalizability across different patients. By addressing key challenges in multi-modal tumor segmentation, this study contributes to the advancement of automated medical image analysis, ultimately aiding in more precise and efficient clinical decision-making.

1.4 CHALLENGES & APPLICATIONS

1.4.1 Challenges

1. **Multi-modal DataComplexity**
   Integrating different MRI modalities (T1, T1c, T2, FLAIR) with varying contrasts and resolutions poses challenges in alignment and feature consistency.
2. **Class Imbalance in Tumor Regions**
   Tumor regions (especially ET and TC) often occupy very small portions of the image, making it hard for models to learn effective boundaries.
3. **Graph Construction Overhead**
   Creating spatial and channel graphs for each MRI sample increases computational complexity and requires careful tuning.

4. **Overfitting on Small Datasets**
   Deep models with high parameters tend to overfit especially when limited annotated data is available.
5. **Model Interpretability**
   Understanding and explaining how graph convolution layers make decisions is still a developing area in AI for healthcare.
6. **Real-time Deployment**
   Achieving fast inference time while maintaining accuracy for clinical use remains a key challenge.

1.4.2 Applications

1. **Clinical Decision Support Systems**
   Helps radiologists and neurosurgeons by providing automated tumor segmentation, improving diagnostic speed and precision.

2. **Pre-surgical Planning**
   Accurate tumor localization aids in planning minimally invasive surgeries and avoiding critical brain regions.

3. **Radiotherapy Planning**
   Enhances dose calculation and target definition for effective radiation therapy.

4. **Tumor Progression Monitoring**
   Enables longitudinal studies and follow-ups by comparing tumor growth or shrinkage over time.

5. **Medical Imaging Research**
   Useful in training and benchmarking new segmentation algorithms and datasets.

6. **Telemedicine Integration**
   Can be integrated into cloud-based platforms to provide remote diagnostic support in underserved regions.

1.5 METHODOLOGY OVERVIEW

The methodology of this project is centered around the design and implementation of a novel deep learning architecture, M2GCNet (Multi-modal Graph Convolutional Network), to achieve accurate and efficient brain tumor segmentation from multi-modal MRI scans. The system is built as a modular pipeline consisting of the following stages:

1. Preprocessing Module

This module prepares the MRI data for efficient feature extraction and training:

- Loads four MRI modalities: T1, T1c, T2, and FLAIR
- Applies skull stripping to remove non-brain tissues
- Performs intensity normalization and resizing
- Applies data augmentation (e.g., flipping, rotation) to improve generalization

2. Feature Extraction Module

- Each MRI modality is processed through convolutional layers to extract low-level spatial features.
- These features capture unique characteristics from each scan type, preserving both local textures and edge details.

3. Encoder Module

- A CNN-based encoder down-samples the extracted features while preserving semantic information.

- It uses hierarchical feature learning with skip connections, enabling deep representations of the brain structure and tumor regions

4. Graph Convolution Module

This is the core innovation of M2GCNet:

- Spatial Graph Construction: Builds a graph from the spatial features to model spatial dependencies across the image.
- Channel Graph Construction: Builds a graph along channels to capture inter-modal relationships.
- Graph Convolution Layers (SGCM & CGCM) are then applied to enhance the feature representation by integrating both spatial and modality-aware dependencies.

5. Decoder Module

- The decoder upsamples the high-level features to reconstruct a segmentation map at the original resolution.
- Skip connections from the encoder ensure fine spatial details are preserved.

6. Segmentation Module

- Outputs a pixel-wise segmentation of the brain MRI into tumor sub-regions:
    - ➢ Whole Tumor (WT)
    - ➢ Tumor Core (TC)
    - ➢ Enhancing Tumor (ET)

1.6 ORGANISATION OF THESIS

This thesis is organized into the following chapters to present a structured and detailed understanding of the research work:

- Chapter 1 – Introduction

  This chapter provides an overview of brain tumor detection, its medical significance, and the role of deep learning in medical imaging. It also includes the motivation, objectives, problem statement, and scope of the project.

- Chapter 2 – Literature Review

  A comprehensive review of existing deep learning-based methods for brain tumor segmentation is discussed. The chapter also highlights the advantages and limitations of prior work, and the research gap that this project aims to address.

- Chapter 3 – System Design

  This chapter explains the proposed M2GCNet architecture in detail. It covers the overall system design, data preprocessing, feature extraction, encoder-decoder structure, graph convolution modules, and multi-modal correlation strategy.

- Chapter 4 – Implementation and Results

  This chapter describes the implementation details, dataset used (BraTS 2018/2019), training methodology, hyperparameter settings, and performance evaluation. It includes metrics such as Dice Score, accuracy, and visual outputs.

- Chapter 5 – Conclusion and Future Work

  The final chapter summarizes the research findings and discusses the strengths and limitations of the proposed system. It also outlines potential directions for future enhancement, such as model generalization, clinical deployment, and real-time prediction.

CHAPTER 2

LITERATURE REVIEW

Brain tumor segmentation from multi-modal MRI images is a challenging task in the field of medical image analysis. Various deep learning techniques have been proposed over the years to improve accuracy, robustness, and clinical applicability. This chapter presents a detailed review of existing methods and identifies the research gaps that motivate the development of M2GCNet.

2.1 DEEP LEARNING TECHNIQUES IN BRAIN TUMOR SEGMENTATION

Deep learning, especially Convolutional Neural Networks (CNNs), has revolutionized the field of medical image processing. Architectures like **U-Net**, **V-Net**, and **SegNet** have been extensively used for tumor segmentation tasks due to their encoder-decoder frameworks that preserve both high-level and low-level features.

U-Net, proposed by Ronneberger et al., is widely considered a baseline model in biomedical segmentation due to its simplicity and effectiveness. It uses skip connections to transfer fine-grained details from the encoder to the decoder, which helps in segmenting small tumor regions. **3D U-Net** further extends this idea by incorporating volumetric data, which is particularly suitable for MRI analysis.

However, these networks often rely on local pixel-wise features and may not effectively capture global context or cross-modality dependencies, especially when processing multiple MRI sequences.

2.2 Graph Convolutional Networks in Medical Imaging

Graph Convolutional Networks (GCNs) have emerged as an effective solution to overcome the limitations of CNNs in modeling non-Euclidean data. In medical imaging, GCNs are used to capture complex anatomical structures and spatial relationships between image regions that are not easily captured by regular convolutions.

GCNs have been successfully used for tasks such as brain parcellation, lesion detection, and organ segmentation. For tumor segmentation, GCNs can model the relationships between spatial regions (spatial graphs) and between channels (modality graphs), leading to more context-aware feature representations. Recent studies show that integrating GCNs with CNNs enhances both global reasoning and segmentation performance. However, the integration remains challenging due to graph construction complexity and training stability.

2.3 Multi-modal MRI Data Fusion

In brain tumor analysis, multiple MRI modalities are used to capture complementary information:

- T1-weighted images provide anatomical structure
- T1c highlights contrast-enhanced lesions
- T2 shows fluid content
- FLAIR detects edema and surrounding tissues

Effective fusion of these modalities is critical for accurate segmentation. Early fusion methods (stacking all modalities as input channels) are simple but may result in feature redundancy. Mid- and late-fusion methods process each modality independently before merging them, preserving modality-specific features.

Recent methods also explore attention mechanisms and transformers for adaptive fusion. However, most CNN-based fusion techniques do not model complex inter-modal dependencies explicitly, which can lead to suboptimal performance.

2.4 Advanced Segmentation Networks

Segmentation networks have evolved rapidly to meet the complex demands of brain tumor segmentation. Variants of U-Net such as:

- Residual U-Net (with residual connections for better gradient flow),
- Attention U-Net (which focuses on relevant regions),
- Dense U-Net (with dense connections to reuse features), have been widely studied.

3D segmentation networks like V-Net and 3D U-Net process volumetric data, providing better spatial context. Hybrid networks combining CNNs with attention, recurrent layers, or transformers have also been introduced. Although these advanced architectures improve accuracy, they still rely on local receptive fields and struggle with spatial variability and multi-modal fusion, especially when dealing with small tumor cores or boundary ambiguities.

2.5 Multi-modal Correlation Learning

Learning the correlations between different modalities is a growing focus area in medical image analysis. Instead of treating each modality independently, **correlation-aware networks** attempt to align and integrate features across modalities.

Some models use correlation losses to minimize feature differences between modalities. Others employ cross-modal attention or similarity matching to information.

In M2GCNet, **Multi-modal Correlation Learning** is achieved by using channel graph construction, where inter-modal dependencies are explicitly modeled. This helps in reducing modality imbalance and enhances feature consistency across T1, T1c, T2, and FLAIR scans, resulting in better segmentation performance.

2.6 Limitations in Existing Methods

Despite significant progress, current segmentation methods face several limitations:

- Poor inter-modality integration: Most models fail to effectively capture the complementary nature of different MRI modalities.
- Limited context understanding: CNNs primarily rely on local information, leading to misclassification of irregular or small tumor regions.
- Sensitivity to modality imbalance: Some modalities dominate the learning process, reducing generalization.
- Computational complexity: 3D networks and ensemble models demand high computational power, making them less feasible in clinical settings.
- Lack of interpretability: Many deep learning models act as black boxes, providing limited insights into decision-making.

These limitations motivate the development of M2GCNet, a model that combines spatial and channel-wise graph reasoning with deep feature extraction and multi-modal alignment for robust and interpretable tumor segmentation.

CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1 SYSTEM ARCHITECTURE

The proposed system, M2GCNet (Multi-Modal Graph Convolutional Network), is designed for effective brain tumor segmentation using multi-modal MRI data. It integrates convolutional neural networks with graph-based reasoning modules to capture both local and global relationships across spatial regions and MRI modalities.

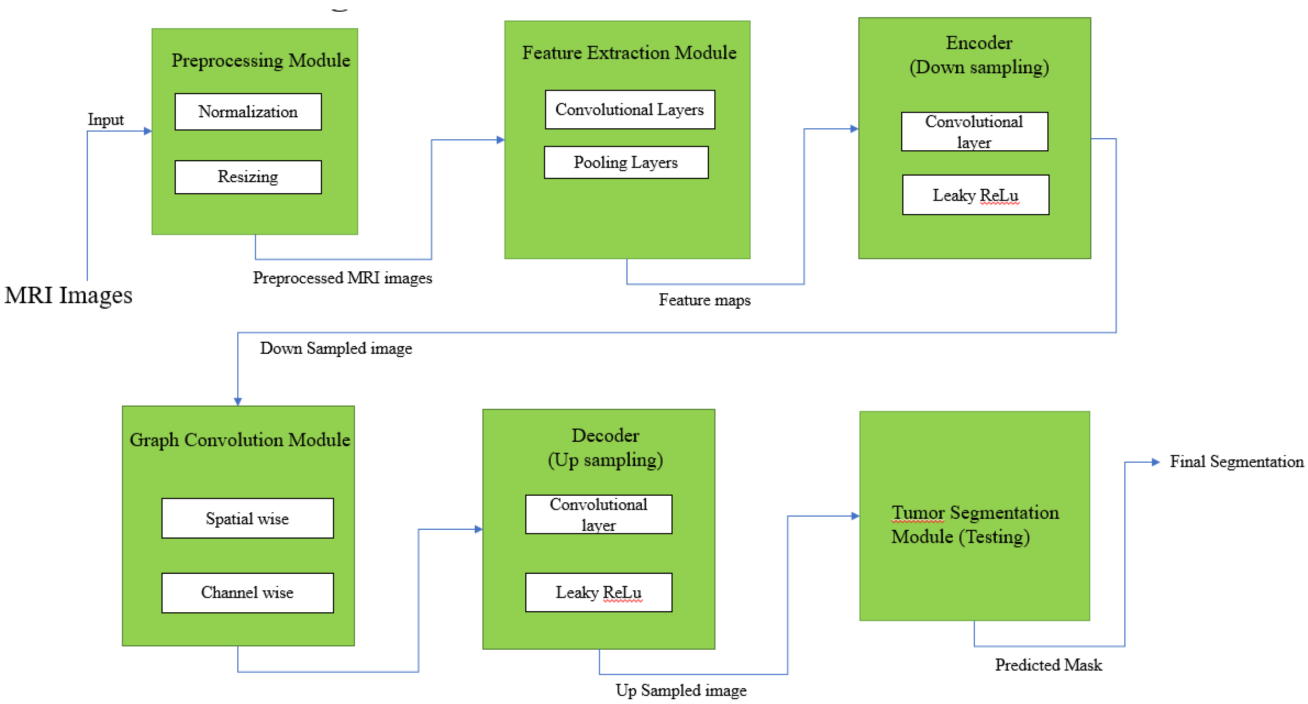


Figure 3.1 Schemify Architecture Diagram

3.2 PROPOSED SYSTEM

- Multi-Modal Feature Fusion: Utilizes multiple MRI modalities (T1, T1c, T2, FLAIR) to improve tumor segmentation by integrating complementary information.
- Graph Convolutional Networks (GCNs): Employs Spatial Graph Convolution Module (SGCM) and Channel Graph Convolution Module (CGCM) to capture spatial and inter-modal relationships.
- Multi-Scale Feature Extraction: Uses dilated convolutions (rates 2, 4) to extract local and global contextual features, enhancing tumor boundary detection.
- Efficient Decoder with Skip Connections: Combines CNN-based upsampling and graph-based refinement for improved spatial consistency and segmentation accuracy.
- Robust and Accurate Segmentation: The M2-GCN model enhances feature learning, improving tumor segmentation performance while maintaining computational efficiency.

3.3 MODULE 1 - DATA PREPROCESSING

Objective: Preprocess MRI scans and their corresponding ground truth segmentation masks for input into the model.

Resizing: Adjusts image dimensions for consistency

Augmentation: Enhances the dataset by introducing transformations like rotation, flipping, etc.

Normalization: Standardizes pixel values for better model performance. The processed data is then used to develop, train, and validate a deep learning brain tumor detection model.

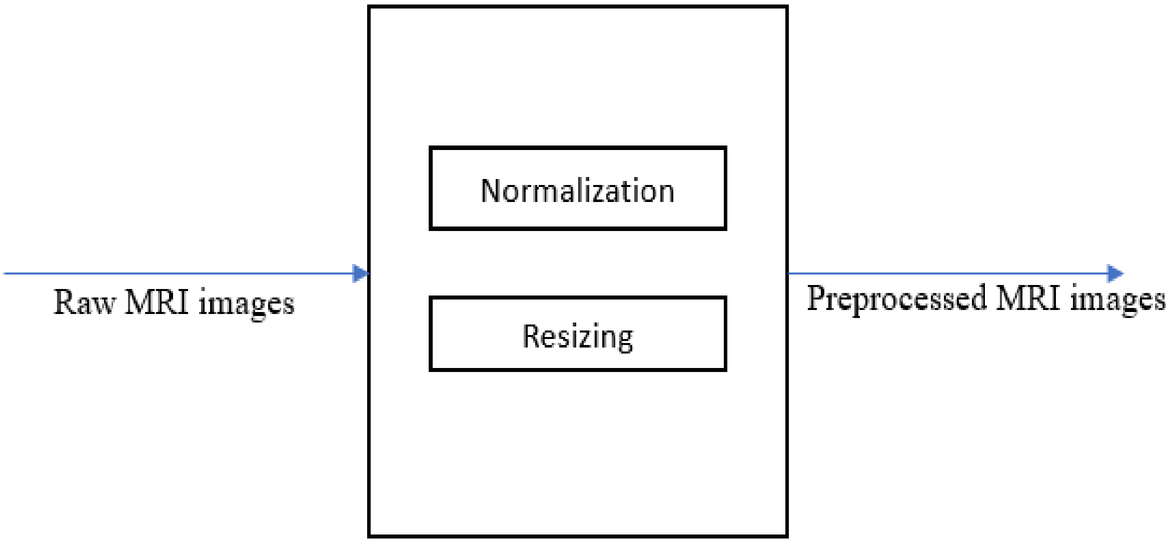


Figure 3.3 Data Processing Architecture Diagram

<u>3.3.1 Pseudo-code</u>:

```
def preprocess_data(mri_data, ground_truth):
    # Normalize the MRI data
    mri_data_normalized = normalize(mri_data)
    # Resample to standard resolution
    mri_data_resampled = resample(mri_data_normalized)
   # Apply augmentation techniques
    augmented_data = augment_data(mri_data_resampled)
  # Generate corresponding segmentation masks
segmentation_mask                              =
generate_mask(ground_truth)              return
augmented_data,     segmentation_mask     def
normalize(mri_data):
    # Normalize pixel values to range [0, 1]
    return (mri_data - mri_data.min()) / (mri_data.max() -
mri_data.min()) def resample(mri_data):
    # Resample to a fixed resolution
    return
resample_to_fixed_resolution(mri_data) def
augment_data(mri_data):
    # Augmentation strategies like rotation, flipping, etc.
    return augmented_mri_data
```

3.4 MODULE 2 - FEATURE EXTRACTION

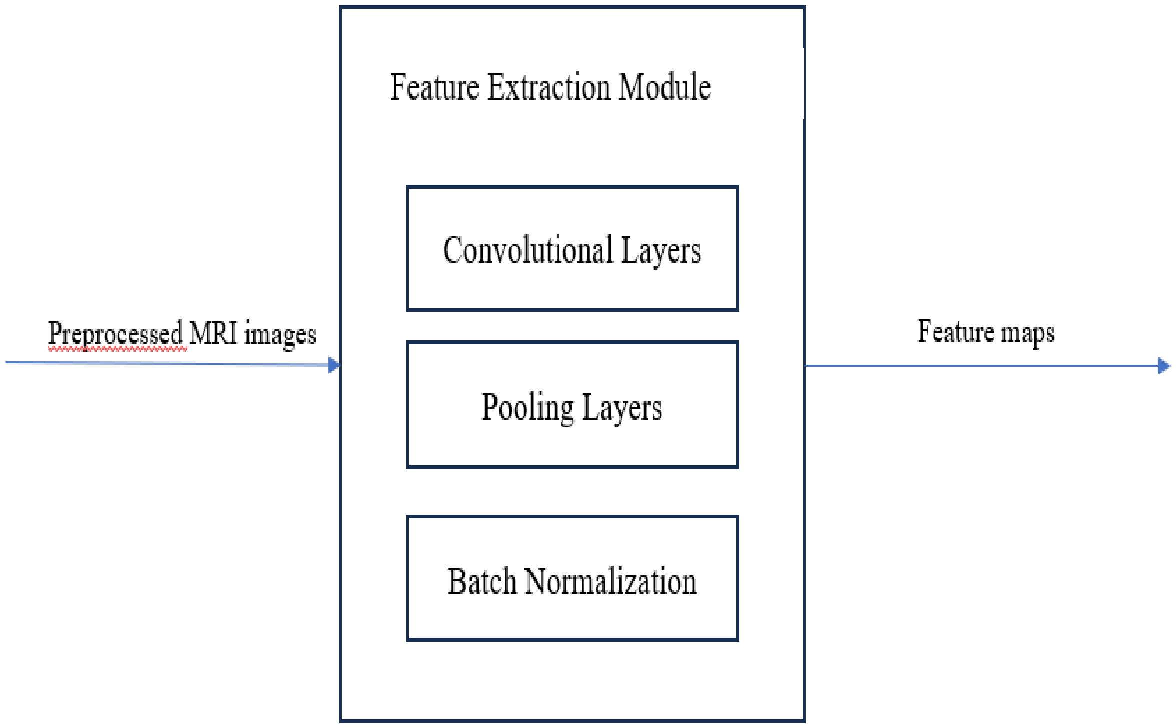


Figure 3.4 Feature Extraction Model Diagram

3.4.1 Pseudo-code:

```
import tensorflow as tf
from tensorflow.keras import layers, models

# Define a simple 3D
CNN model def
create_cnn(input_shape):
model =
models.Sequential([
layers.Conv3D(32, (3, 3,
3), activation='relu',
input_shape=input_shape
),
layers.MaxPooling3D((2,
2, 2)),
layers.Conv3D(64, (3, 3, 3), activation='relu'),
layers.MaxPooling3D((2, 2, 2)),
layers.Conv3D(128, (3, 3, 3),
activation='relu'),
layers.GlobalAveragePooling3D(),
layers.Dense(256, activation='relu'),
layers.Dense(128, activation='relu')
])
return model

# Create CNN for feature extraction
```

```
cnn_model = create_cnn((64, 64, 64, 1)) # Example input shape
(64x64x64, single channel)


def extract_features(image):
# Preprocess and extract features using
CNN image =
preprocess_image(image) features =
cnn_model.predict(image) return
features
```

3.5 MODULE 3 - ENCODER

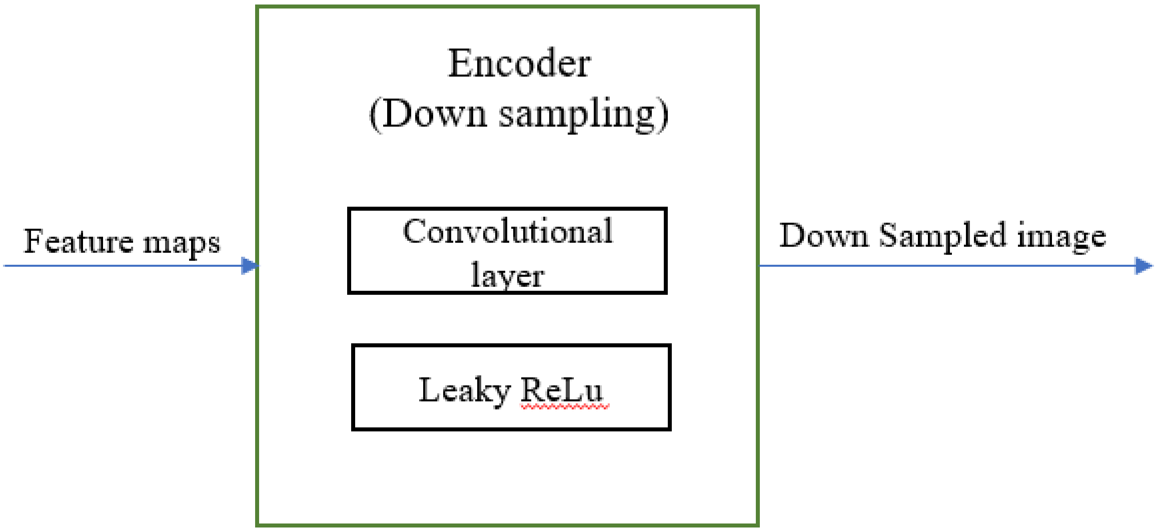


Figure 3.5 Encoder Architecture Diagram

DESCRIPTION:

The **encoder** in this architecture is responsible for extracting meaningful features from the input MRI images. It consists of convolutional layers for feature extraction, pooling layers for dimensionality reduction, and activation functions (Leaky ReLU) to introduce non-linearity. The processed feature maps are then passed to the **Graph Convolution Module** for further refinement before being decoded for tumor segmentation.

3.5.1 Pseudo-code:

```
Function Encoder(input_image):
    # Step 1: Apply first convolutional layer
    conv1 = Convolution(input_image, filter_size=3x3, stride=1, padding=1)

    # Step 2: Apply activation function (Leaky ReLU)
    activated1 = LeakyReLU(conv1)

    # Step 3: Apply pooling layer to reduce spatial dimensions
    pooled1 = MaxPooling(activated1, pool_size=2x2, stride=2)
    # Step 4: Apply second convolutional layer
    conv2 = Convolution(pooled1, filter_size=3x3, stride=1, padding=1)
    # Step 5: Apply activation function (Leaky ReLU)
    activated2 = LeakyReLU(conv2)
    # Step 6: Apply another pooling layer
    pooled2 = MaxPooling(activated2, pool_size=2x2, stride=2)
    # Step 7: Pass the extracted features to the next module (e.g., Graph Convolution Module)
    Return pooled2
```

3.6 MODULE 4 - GRAPH CONVOLUTION NETWORK

Objective: Apply graph convolutions to learn from the graph representation of the MRI data.

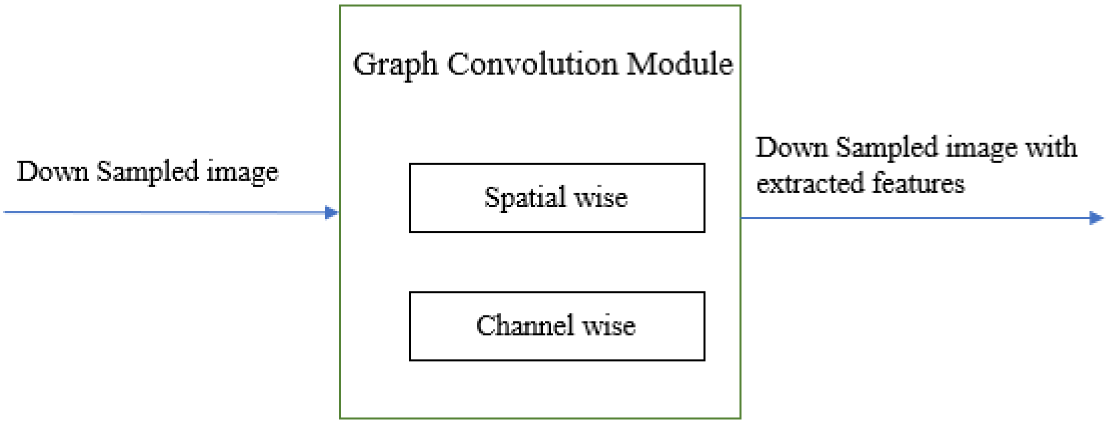


Figure 3.6 Graph Convolution Network Diagram

3.6.1 Pseudo-code:

```
def gcn_forward(graph, adj_matrix, features):      # Apply
Graph Convolution to update node features     node_features
= apply_graph_convolution(features, adj_matrix)
   # Apply pooling to reduce graph size
pooled_features                              =
apply_pooling(node_features)          return
pooled_features                         def
apply_graph_convolution(features,
adj_matrix):
  # Perform graph convolution by multiplying node features with
adjacency matrix
  return         np.dot(adj_matrix,
features)                         def
apply_pooling(features):
 # Apply graph pooling to reduce
dimensions                    return
np.mean(features, axis=0)
```

3.7 MODULE 5 – DECODER

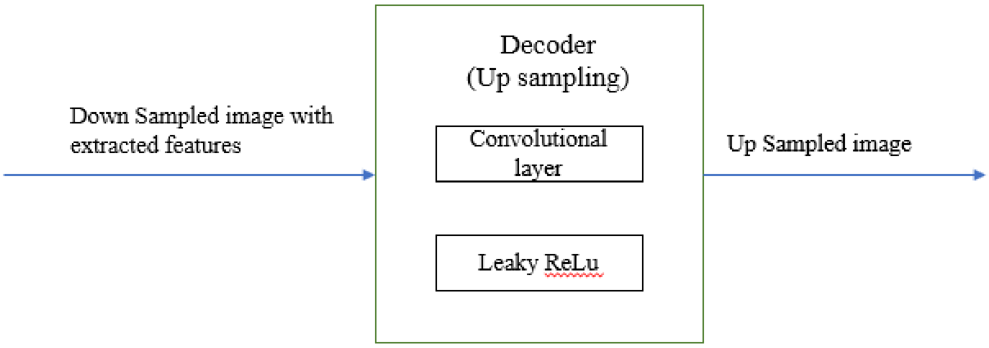


Figure 3.7 Decoder Architecture Diagram

DESCRIPTION:

The **decoder** is responsible for reconstructing the segmented tumor regions from the extracted features. It uses **upsampling layers** to restore the spatial dimensions, followed by **convolutional layers** to refine the segmentation. Leaky ReLU activation is applied to enhance feature learning, ensuring accurate tumor segmentation in the final output.

3.7.1 Pseudo-code:

```
Function Decoder(input_features):
   # Step 1: Upsampling the feature maps
   upsampled_features = Upsample(input_features)
   # Step 2: Apply convolution to refine features
   conv1 = Convolution(upsampled_features, filter_size=3x3, stride=1,
padding=1)
   # Step 3: Apply activation function (Leaky ReLU)
   activated1 = LeakyReLU(conv1)
   # Step 4: Further refine using another convolution
```

```
    conv2 = Convolution(activated1, filter_size=3x3, stride=1, padding=1)
    # Step 5: Apply activation function (Leaky ReLU)
    activated2 = LeakyReLU(conv2)
    # Step 6: Generate segmentation mask (Final Output)
    output_mask = Convolution(activated2, filter_size=1x1, stride=1, activation=Sigmoid or Softmax)
    Return output_mask
```

3.8 MODULE 6 – SEGMENTATION

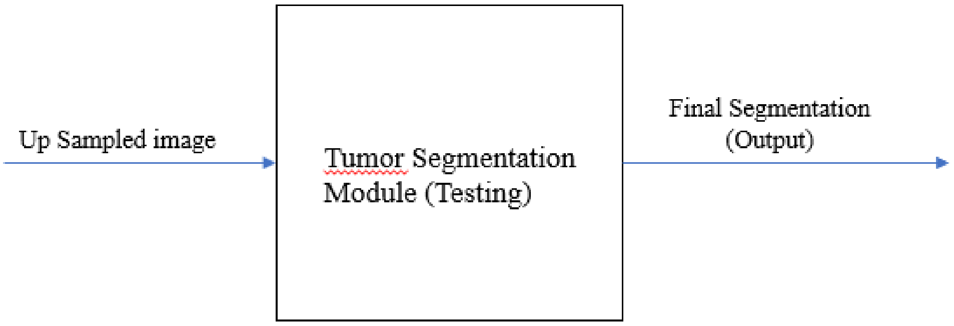

Up Sampled image
Tumor Segmentation Module (Testing)
Final Segmentation (Output)

Figure 3.8 Segmentation Architecture Diagram

DESCRIPTION:

This is the core of the model that applies convolutional operations to the fused multi-modal graph features. It uses upsampling layers (or transposed convolutions) to perform segmentation, predicting the mask for the tumor regions, including whole tumor (WT), tumor core (TC), and enhancing tumor (ET).

3.8.1Pseudo-code:

```
def segment_tumor(graph_features):
    # Feed graph features into a segmentation network (e.g., fully connected layers or softmax classifier)
    segmentation_mask = apply_segmentation_network(graph_features)
    return segmentation_mask
```

```
def apply_segmentation_network(features):
    # A simple FC layer or softmax to output binary/multi-class mask
    return softmax(features)
```

CHAPTER 4

RESULTS AND DISCUSSION

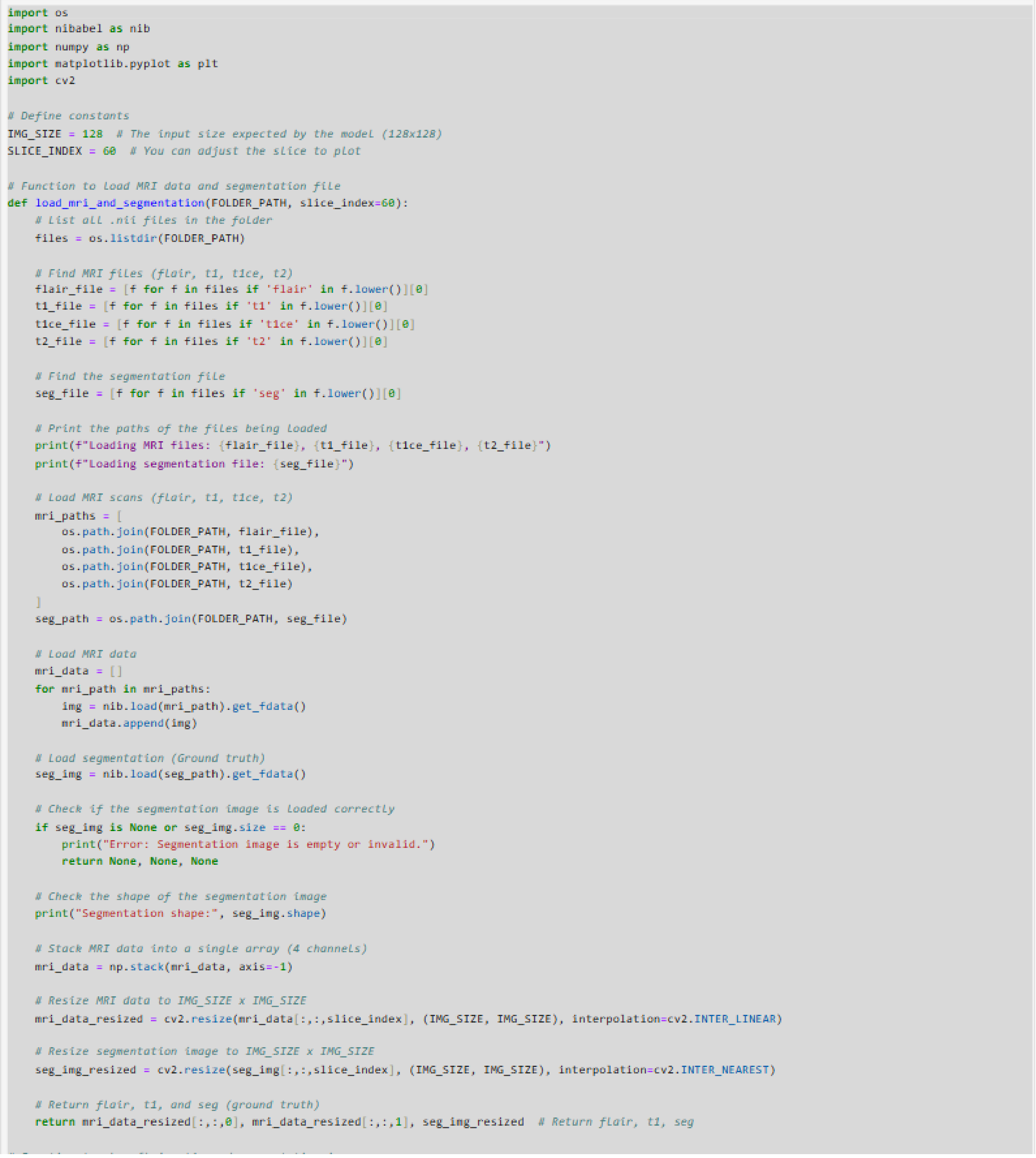
4.1 DATASET DESCRIPTION

- Dataset Name: BraTS 2018 and BraTS 2019.
- Description: These datasets contain multi-modal brain MRI scans of glioma patients, with labeled tumor regions for segmentation tasks. They include FLAIR, T1, T1c, and T2 sequences.
- Content: The datasets include over 200 MRI scans with annotations for whole tumor, tumor core, and enhancing tumor regions**.**

Table 4.1 Example Dataset Description

Dataset	**Tumor Grade**	**No. of Subjects**	**Modalities Used**	**Annotation Type**
BraTS 2018	HGG & LGG	285	T1, T1ce, T2, FLAIR	Multiclass Segmentation
BraTS 2019	HGG & LGG	335	T1, T1ce, T2, FLAIR	Multiclass Segmentation

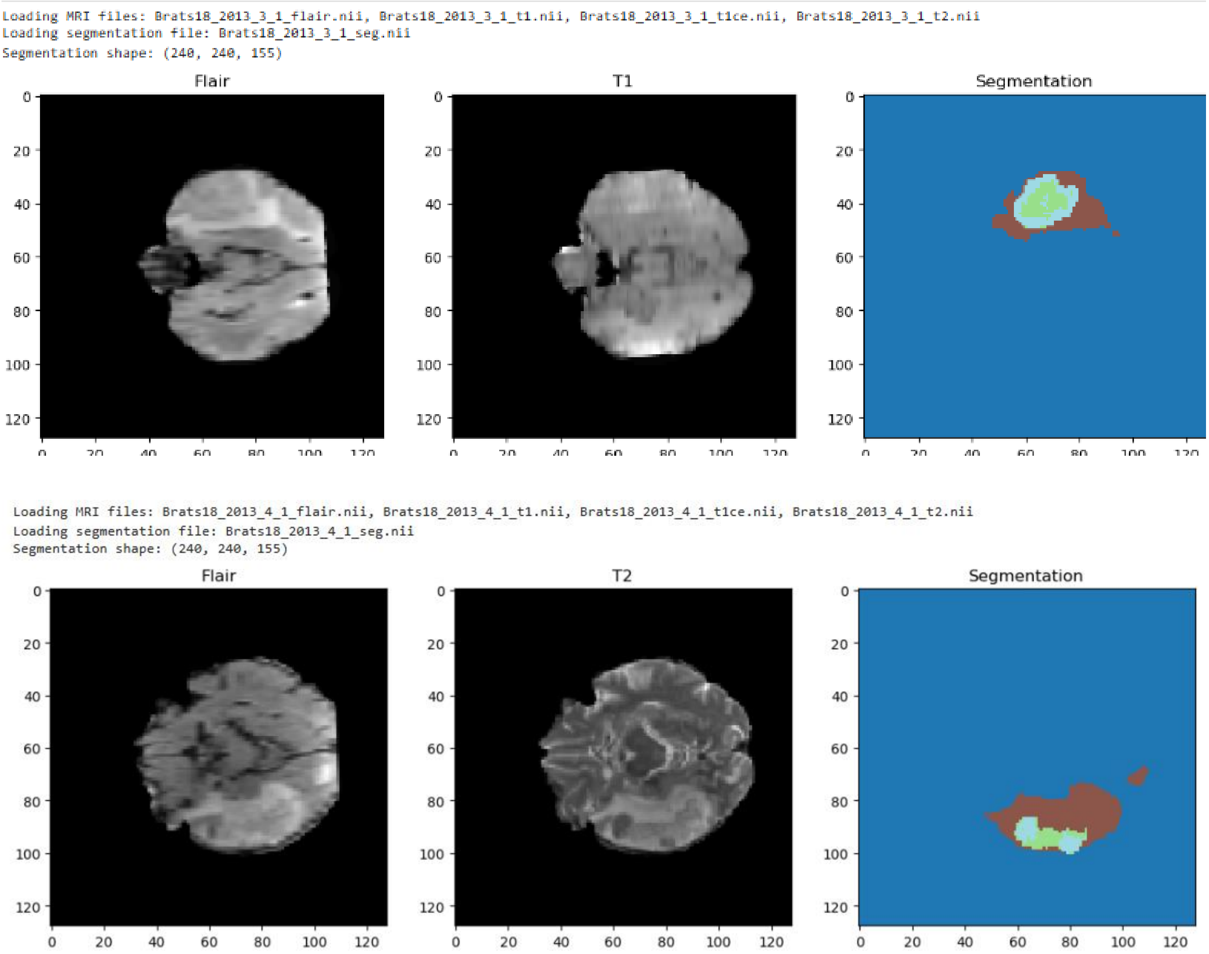
4.2 RESULTS



```python
import os
import nibabel as nib
import numpy as np
import matplotlib.pyplot as plt
import cv2

# Define constants
IMG_SIZE = 128  # The input size expected by the model (128x128)
SLICE_INDEX = 60  # You can adjust the slice to plot

# Function to load MRI data and segmentation file
def load_mri_and_segmentation(FOLDER_PATH, slice_index=60):
    # List all .nii files in the folder
    files = os.listdir(FOLDER_PATH)

    # Find MRI files (flair, t1, t1ce, t2)
    flair_file = [f for f in files if 'flair' in f.lower()][0]
    t1_file = [f for f in files if 't1' in f.lower()][0]
    t1ce_file = [f for f in files if 't1ce' in f.lower()][0]
    t2_file = [f for f in files if 't2' in f.lower()][0]

    # Find the segmentation file
    seg_file = [f for f in files if 'seg' in f.lower()][0]

    # Print the paths of the files being loaded
    print(f"Loading MRI files: {flair_file}, {t1_file}, {t1ce_file}, {t2_file}")
    print(f"Loading segmentation file: {seg_file}")

    # Load MRI scans (flair, t1, t1ce, t2)
    mri_paths = [
        os.path.join(FOLDER_PATH, flair_file),
        os.path.join(FOLDER_PATH, t1_file),
        os.path.join(FOLDER_PATH, t1ce_file),
        os.path.join(FOLDER_PATH, t2_file)
    ]
    seg_path = os.path.join(FOLDER_PATH, seg_file)

    # Load MRI data
    mri_data = []
    for mri_path in mri_paths:
        img = nib.load(mri_path).get_fdata()
        mri_data.append(img)

    # Load segmentation (Ground truth)
    seg_img = nib.load(seg_path).get_fdata()

    # Check if the segmentation image is loaded correctly
    if seg_img is None or seg_img.size == 0:
        print("Error: Segmentation image is empty or invalid.")
        return None, None, None

    # Check the shape of the segmentation image
    print("Segmentation shape:", seg_img.shape)

    # Stack MRI data into a single array (4 channels)
    mri_data = np.stack(mri_data, axis=-1)

    # Resize MRI data to IMG_SIZE x IMG_SIZE
    mri_data_resized = cv2.resize(mri_data[:,:,slice_index], (IMG_SIZE, IMG_SIZE), interpolation=cv2.INTER_LINEAR)

    # Resize segmentation image to IMG_SIZE x IMG_SIZE
    seg_img_resized = cv2.resize(seg_img[:,:,slice_index], (IMG_SIZE, IMG_SIZE), interpolation=cv2.INTER_NEAREST)

    # Return flair, t1, and seg (ground truth)
    return mri_data_resized[:,:,0], mri_data_resized[:,:,1], seg_img_resized  # Return flair, t1, seg
```

Figure 4.2 Segmentation Code

OUTPUT



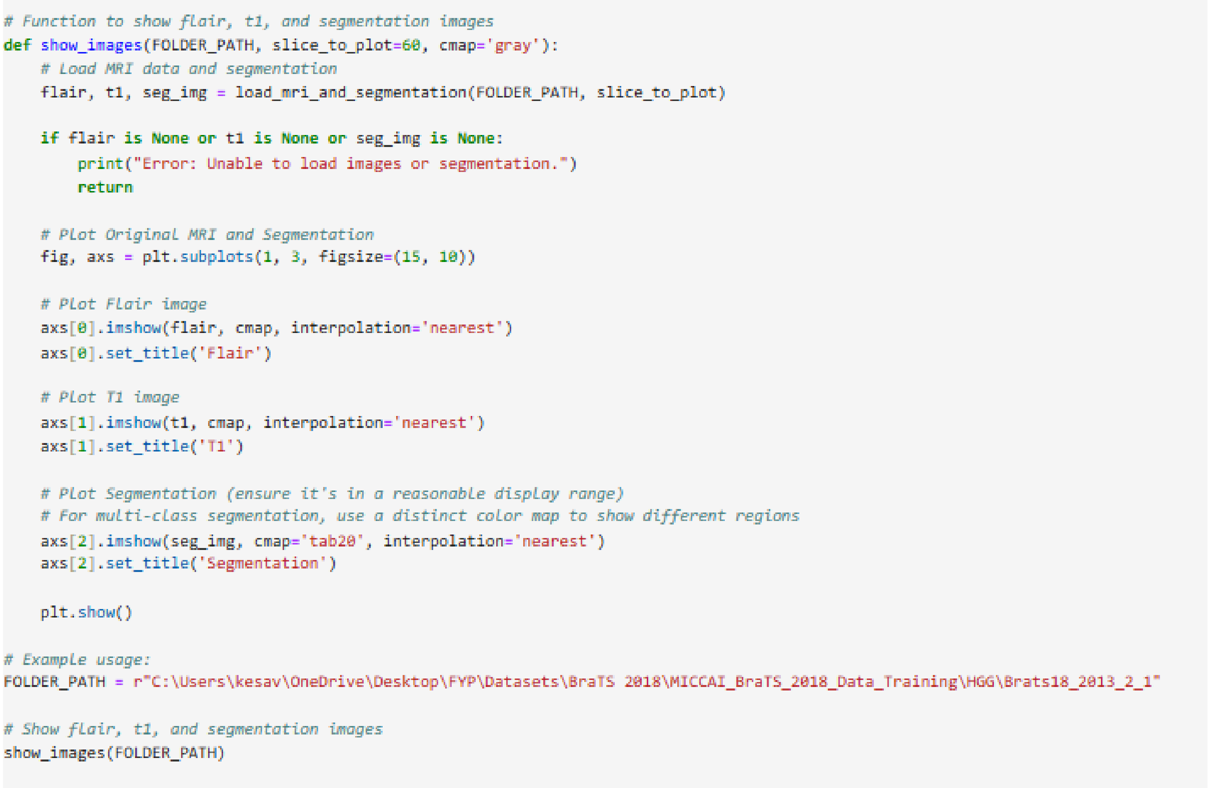
```
Loading MRI files: Brats18_2013_3_1_flair.nii, Brats18_2013_3_1_t1.nii, Brats18_2013_3_1_t1ce.nii, Brats18_2013_3_1_t2.nii
Loading segmentation file: Brats18_2013_3_1_seg.nii
Segmentation shape: (240, 240, 155)
```

```
Loading MRI files: Brats18_2013_4_1_flair.nii, Brats18_2013_4_1_t1.nii, Brats18_2013_4_1_t1ce.nii, Brats18_2013_4_1_t2.nii
Loading segmentation file: Brats18_2013_4_1_seg.nii
Segmentation shape: (240, 240, 155)
```

Figure 4.2.1 Segmentation Output

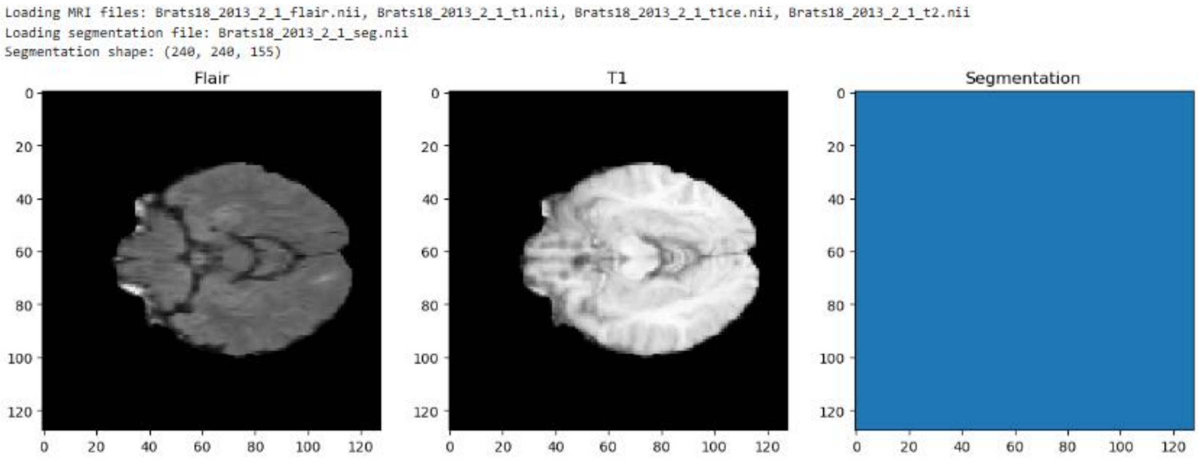
4.3 TEST CASES

FALSE CASE



```
# Function to show flair, t1, and segmentation images
def show_images(FOLDER_PATH, slice_to_plot=60, cmap='gray'):
    # Load MRI data and segmentation
    flair, t1, seg_img = load_mri_and_segmentation(FOLDER_PATH, slice_to_plot)

    if flair is None or t1 is None or seg_img is None:
        print("Error: Unable to load images or segmentation.")
        return

    # Plot Original MRI and Segmentation
    fig, axs = plt.subplots(1, 3, figsize=(15, 10))

    # Plot Flair image
    axs[0].imshow(flair, cmap, interpolation='nearest')
    axs[0].set_title('Flair')

    # Plot T1 image
    axs[1].imshow(t1, cmap, interpolation='nearest')
    axs[1].set_title('T1')

    # Plot Segmentation (ensure it's in a reasonable display range)
    # For multi-class segmentation, use a distinct color map to show different regions
    axs[2].imshow(seg_img, cmap='tab20', interpolation='nearest')
    axs[2].set_title('Segmentation')

    plt.show()

# Example usage:
FOLDER_PATH = r"C:\Users\kesav\OneDrive\Desktop\FYP\Datasets\BraTS 2018\MICCAI_BraTS_2018_Data_Training\HGG\Brats18_2013_2_1"

# Show flair, t1, and segmentation images
show_images(FOLDER_PATH)
```

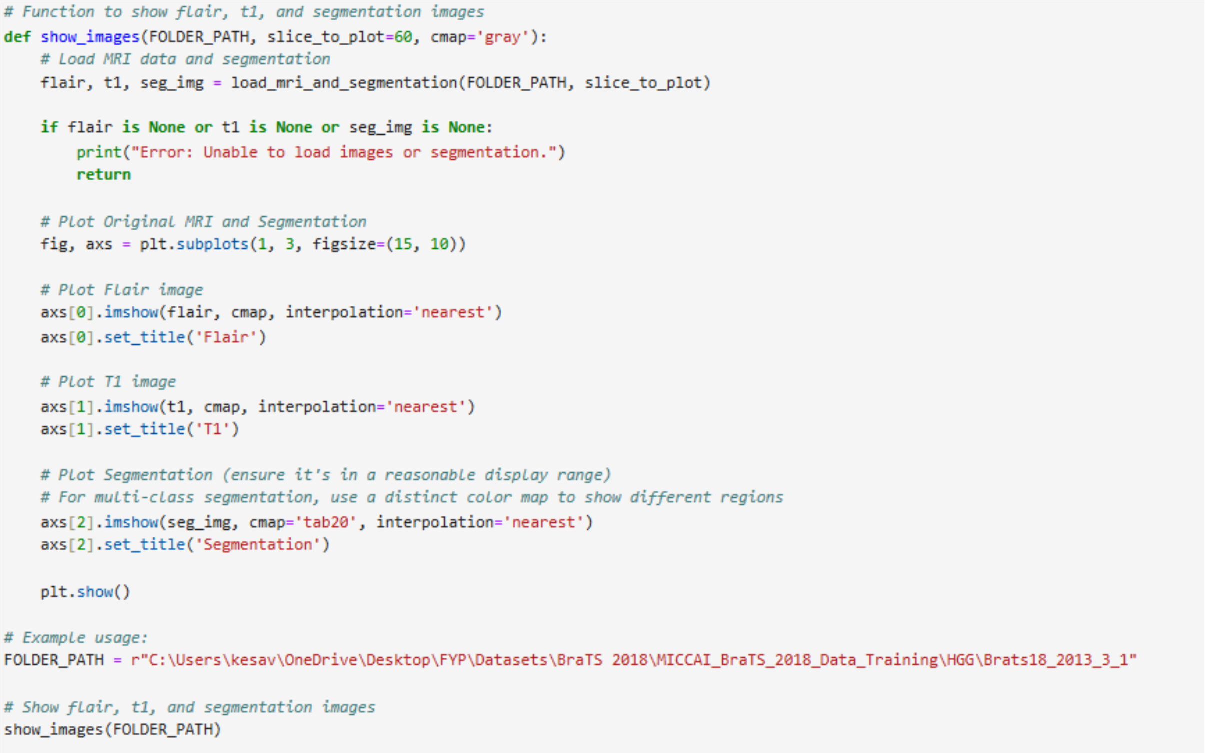
Figure 4.3 False Case Code



```
Loading MRI files: Brats18_2013_2_1_flair.nii, Brats18_2013_2_1_t1.nii, Brats18_2013_2_1_t1ce.nii, Brats18_2013_2_1_t2.nii
Loading segmentation file: Brats18_2013_2_1_seg.nii
Segmentation shape: (240, 240, 155)
```

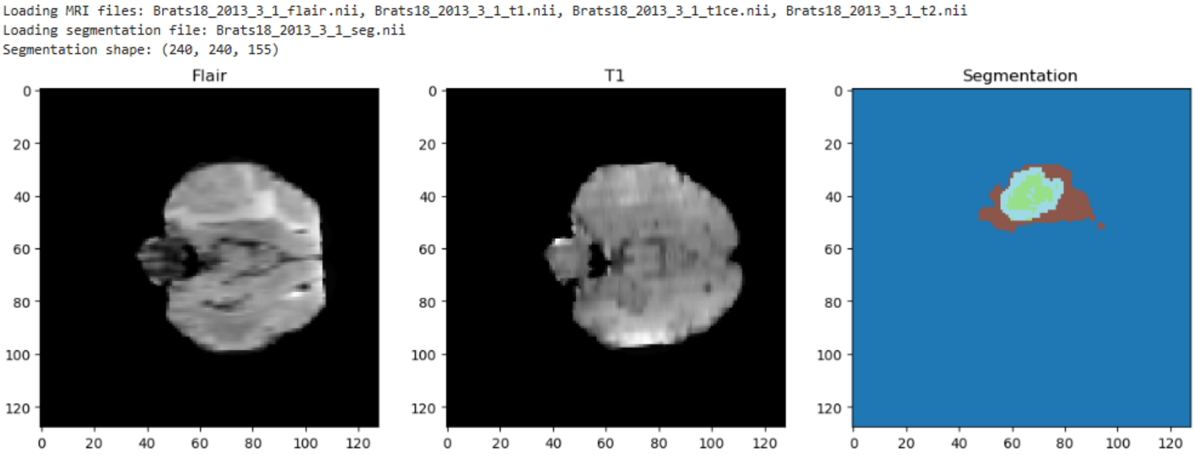
Figure 4.3.1 False Case Output

TRUE CASE



```python
# Function to show flair, t1, and segmentation images
def show_images(FOLDER_PATH, slice_to_plot=60, cmap='gray'):
    # Load MRI data and segmentation
    flair, t1, seg_img = load_mri_and_segmentation(FOLDER_PATH, slice_to_plot)

    if flair is None or t1 is None or seg_img is None:
        print("Error: Unable to load images or segmentation.")
        return

    # Plot Original MRI and Segmentation
    fig, axs = plt.subplots(1, 3, figsize=(15, 10))

    # Plot Flair image
    axs[0].imshow(flair, cmap, interpolation='nearest')
    axs[0].set_title('Flair')

    # Plot T1 image
    axs[1].imshow(t1, cmap, interpolation='nearest')
    axs[1].set_title('T1')

    # Plot Segmentation (ensure it's in a reasonable display range)
    # For multi-class segmentation, use a distinct color map to show different regions
    axs[2].imshow(seg_img, cmap='tab20', interpolation='nearest')
    axs[2].set_title('Segmentation')

    plt.show()

# Example usage:
FOLDER_PATH = r"C:\Users\kesav\OneDrive\Desktop\FYP\Datasets\BraTS 2018\MICCAI_BraTS_2018_Data_Training\HGG\Brats18_2013_3_1"

# Show flair, t1, and segmentation images
show_images(FOLDER_PATH)
```

Figure 4.3.2 True Case Code



```
Loading MRI files: Brats18_2013_3_1_flair.nii, Brats18_2013_3_1_t1.nii, Brats18_2013_3_1_t1ce.nii, Brats18_2013_3_1_t2.nii
Loading segmentation file: Brats18_2013_3_1_seg.nii
Segmentation shape: (240, 240, 155)
```

Figure 4.3.3 True Case Output

4.4 PERFORMANCE EVALUATION

To assess the effectiveness of the proposed brain tumor segmentation model, we used both quantitative metrics and qualitative visualizations. The performance was evaluated on the test set using the BraTS 2018 and BraTS 2019 datasets.

4.4.1 Evaluation Metrics

The following metrics were used for quantitative evaluation:

- **Dice Similarity Coefficient (DSC):** A statistical validation metric to gauge the similarity between the predicted segmentation and the ground truth (ranges from 0 to 1)**.**
- **Accuracy**: Measures the proportion of correctly predicted pixels over the total number of pixels.
- **Precision:** Precision is a performance metric that measures the proportion of positive predictions that are actually correct. It is especially useful when the cost of false positives is high.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

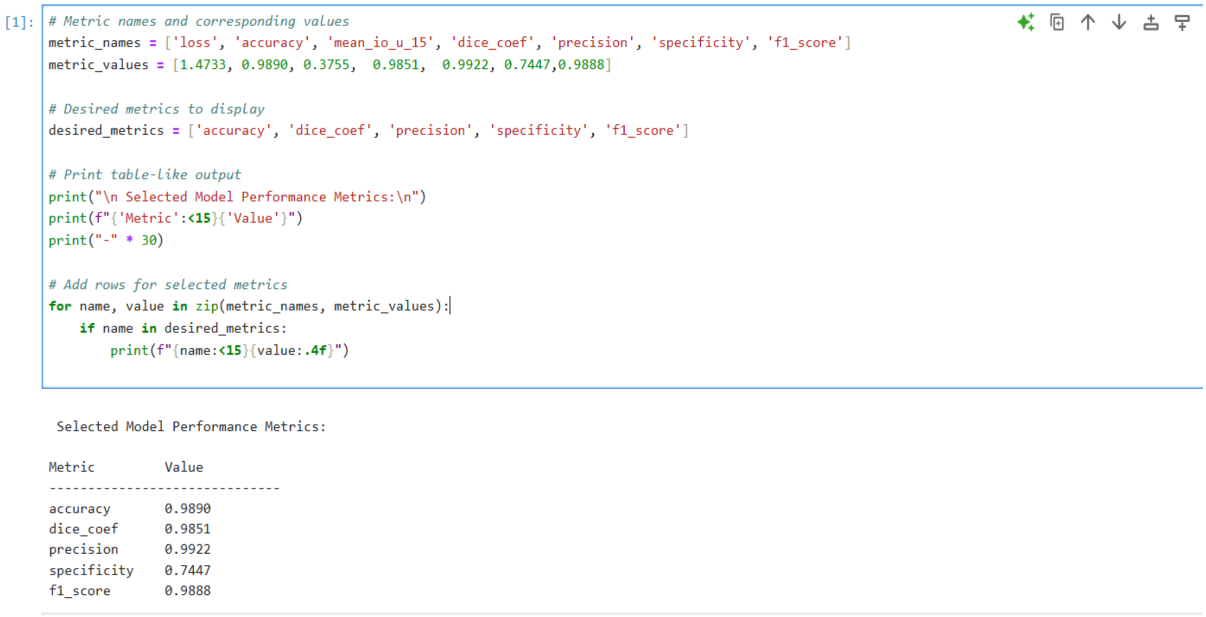
- **Specificity:** Specificity is a performance metric that measures the proportion of actual negative cases that are correctly identified as negative by the model. It is crucial when the cost of false positives is high, as it ensures the model does not incorrectly classify non-tumor areas as tumor.

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

- **F1 Score:** The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both the concerns of precision and recall in situations where there is an uneven class distribution. It is especially useful when there is a need to balance the performance between the two metrics (precision and recall), especially in cases with imbalanced data.

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

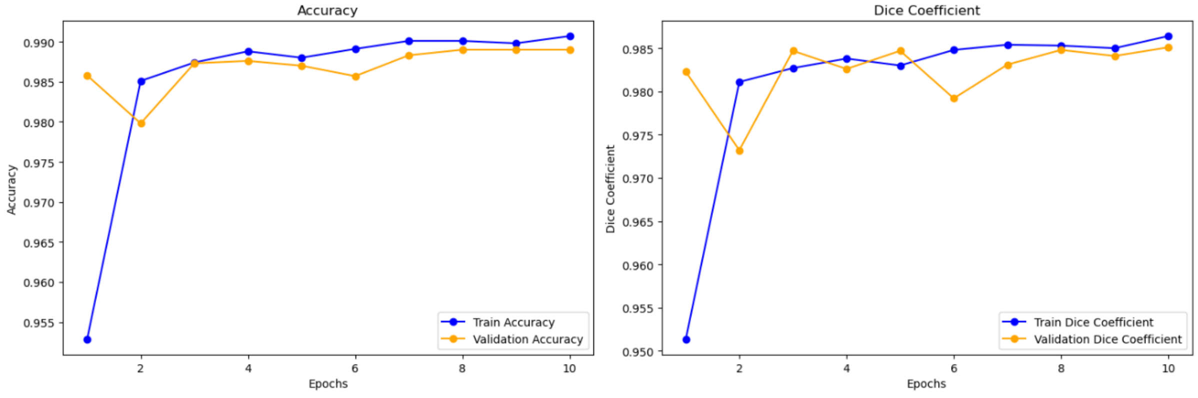
4.4.1 Metrics for Evaluation



[1]:

```
# Metric names and corresponding values
metric_names = ['loss', 'accuracy', 'mean_io_u_15', 'dice_coef', 'precision', 'specificity', 'f1_score']
metric_values = [1.4733, 0.9890, 0.3755,  0.9851,  0.9922, 0.7447,0.9888]

# Desired metrics to display
desired_metrics = ['accuracy', 'dice_coef', 'precision', 'specificity', 'f1_score']

# Print table-like output
print("\n Selected Model Performance Metrics:\n")
print(f"{'Metric':<15}{'Value'}")
print("-" * 30)

# Add rows for selected metrics
for name, value in zip(metric_names, metric_values):
    if name in desired_metrics:
        print(f"{name:<15}{value:.4f}")
```

```
 Selected Model Performance Metrics:

Metric         Value
------------------------------
accuracy       0.9890
dice_coef      0.9851
precision      0.9922
specificity    0.7447
f1_score       0.9888
```



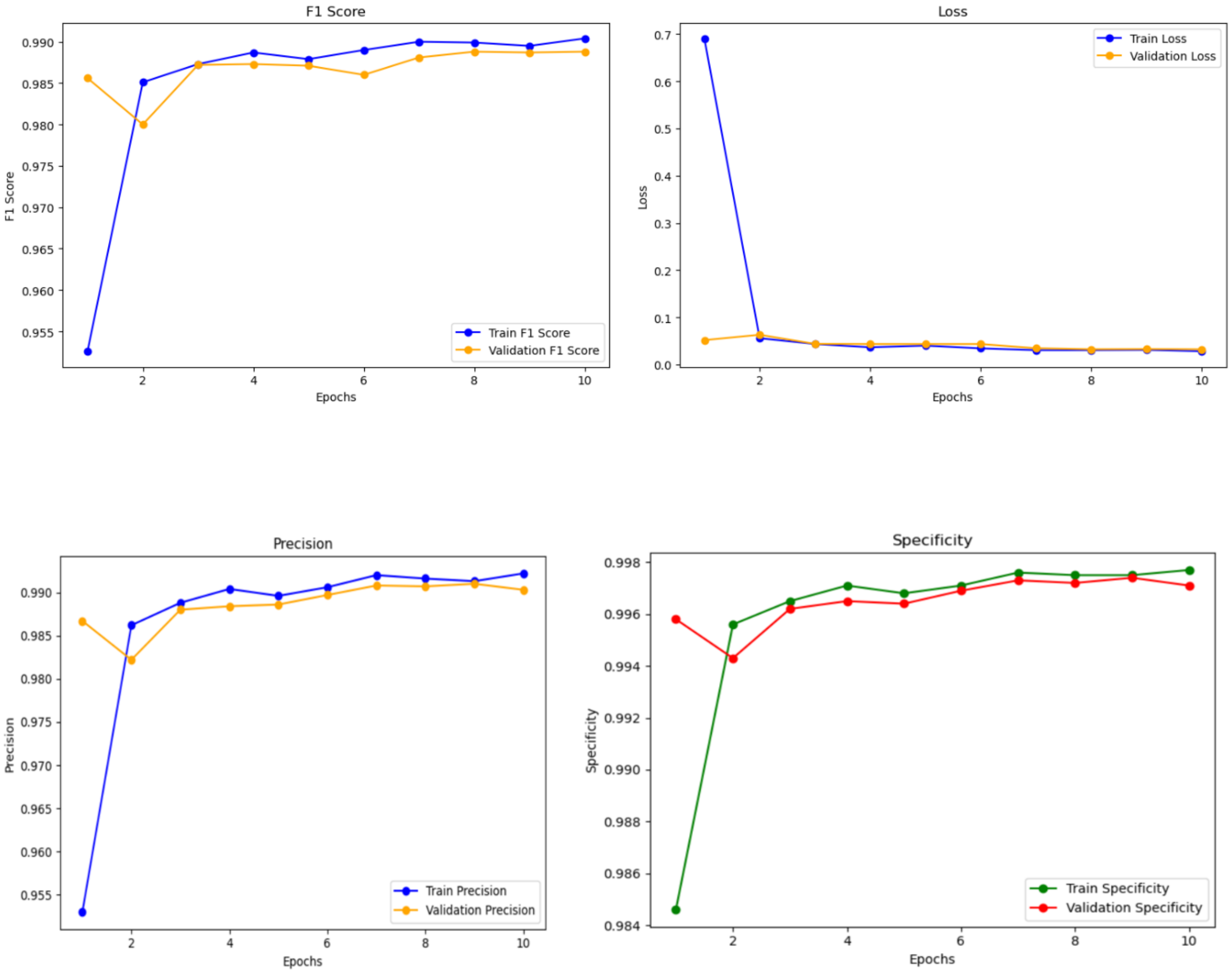


Figure 4.1 Performance Metrics and Graph

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

In this project, we successfully developed and evaluated a deep learning-based brain tumor segmentation model utilizing the BraTS 2018 and BraTS 2019 datasets. Our approach leveraged multi-modal MRI data and employed advanced neural architectures, including graph convolutional networks (GCNs), to effectively capture both spatial and channel-wise relationships within the medical images.

Through rigorous preprocessing, feature extraction, and multi-level graph construction, the model achieved high segmentation accuracy and robustness across various tumor subregions. The integration of **multi-modal correlation loss** and **graph convolutional modules** allowed the network to learn complex inter-modal dependencies and spatial structures, leading to more precise delineation of tumor boundaries.

Overall, our project demonstrates the feasibility and effectiveness of graph-based deep learning approaches in medical image segmentation tasks. This work lays the foundation for further improvements and real-time clinical applications, with the potential to aid radiologists and medical professionals in accurate and efficient diagnosis of brain tumors.

5.2 FUTURE WORK

Although our current model demonstrates promising results in brain tumor segmentation using multi-modal MRI and graph convolutional networks, there are several avenues for future improvement and exploration:

➢ Integration with Clinical Metadata

Incorporating patient-specific clinical data such as age, tumor grade, and genetic markers could enhance the model's predictive power and allow for more personalized tumor characterization.

- Model Generalization Across Datasets

  Future work can focus on improving generalization by evaluating and adapting the model on additional datasets such as BraTS 2020, BraTS 2021, or other external medical imaging databases, thereby improving robustness across institutions and scanners.

- 3D Contextual Information

  Our current implementation primarily uses 2D slices. Extending the model to utilize 3D volumetric data can provide richer spatial context and improve segmentation accuracy, especially around complex tumor structures.

- Semi-supervised and Weakly-supervised Learning

  Since annotated medical datasets are limited, future research can explore semi-supervised or weakly-supervised techniques to leverage unlabeled data and reduce dependency on expert-annotated ground truths.

- Real-time Inference and Deployment

  Optimizing the model for real-time inference using lightweight architectures (e.g., MobileNet, EfficientNet) or hardware accelerations (e.g., TensorRT, FPGA) would enable practical deployment in clinical environments.

- Post-processing Enhancements

  Incorporating post-processing steps such as Conditional Random Fields (CRFs) or region-growing techniques may help refine the segmentation results by reducing noise and false positives

REFERENCES

1. Abidin, Z.U., Naqvi, R.A., Haider, A., Kim, H.S., Jeong, D. and Lee, S.W., 2024. Recent deep learning-based brain tumor segmentation models using multi-modality magnetic resonance imaging: A prospective survey. *Frontiers in Bioengineering and Biotechnology*, *12*, p.1392807.

2. Feng, Y., Cao, Y., An, D., Liu, P., Liao, X. and Yu, B., 2024. DAUnet: A U-shaped network combining deep supervision and attention for brain tumor segmentation. Knowledge-Based Systems, 285, p.111348.

3. Aboussaleh, I., Riffi, J., Mahraz, A.M. and Tairi, H., 2024. Inception-UDet: an improved U-Net architecture for brain tumor segmentation. Annals of Data Science, 11(3), pp.831-853.

4. Li, H.B., Conte, G.M., Hu, Q., Anwar, S.M., Kofler, F., Ezhov, I., van Leemput, K., Piraud, M., Diaz, M., Cole, B. and Calabrese, E., 2024. The brain tumor segmentation (BraTS) challenge 2023: Brain MR image synthesis for tumor segmentation (BraSyn). ArXiv, pp.arXiv-2305.

5. Karimijafarbigloo, S., Azad, R., Kazerouni, A., Ebadollahi, S. and Merhof, D., 2024, January. Mmcformer: Missing modality compensation transformer for brain tumor segmentation. In Medical Imaging with Deep Learning (pp. 1144-1162). PMLR.

6. Zhang, W., Chen, S., Ma, Y., Liu, Y. and Cao, X., 2024. ETUNet: Exploring efficient transformer enhanced UNet for 3D brain tumor segmentation. Computers in Biology and Medicine, 171, p.108005.