Project Report

On

# An Ensemble Model for Glioma Sub-Region Segmentation With CNN's

Thesis submitted in partial fulfillment of the requirements for the award of degree of

**Bachelor of Engineering**

In

**Computer Science and Engineering**

By

|  |  |
|---|---|
| **Alugula Yugendhar Reddy** | **1608-20-733-062** |
| **Pabba Akhil** | **1608-20-733-084** |
| **Aerpula Arundhathi** | **1608-20-733-109** |

Under the guidance of

**Mrs. K Bhagya Laxmi**

**Assistant Professor**

**Department of Computer Science and Engineering**

**Matrusri Engineering College**

**Accredited by NBA & NAAC**

(Affiliated to Osmania University, Approved by AICTE)

Saidabad, Hyderabad-500059

2023-2024

**Department of Computer Science and Engineering**

**Matrusri Engineering College**

**Accredited by NBA & NAAC**

(Affiliated to Osmania University, Approved by AICTE)
Saidabad, Hyderabad-500059
2023-2024

## CERTIFICATE

This is to Certify that a Project report entitled **"An Ensemble Model For Glioma Sub Region Segmentation With CNN's"** is being submitted by Alugula Yugendhar Reddy **(**1608-20-733-062**),** Pabba Akhil **(**1608-20-733-084**),** and Aerpula Arundhathi **(** 1608-20-733-109**)** in partial fulfillment of the requirement of the award for the degree of Bachelor of Engineering in "Computer Science and Engineering" O.U., Hyderabad during the year 2023-2024 is a record of bonafide work carried out by them under my guidance. The results presented in this thesis have been verified and are found to be satisfactory.

**Project Guide** **H.O.D.**

**Mrs . K Bhagya Laxmi**
**Assistant Professor,**
**Dept of C.S.E.**

**Dr P Vijaya Pal Reddy**
**Professor & Head,**
**Dept. of CSE**

**External Examiner(s)**

**Matrusri Engineering College**
**Accredited by NBA & NAAC**
(Affiliated to Osmania University, Approved by AICTE)
Saidabad, Hyderabad-500059
**Department of Computer Science and Engineering**

## DECLARATION BY THE CANDIDATE

We, **Alugula Yugendhar Reddy** bearing **H.T.No:1608-20-733-062**, **Pabba Akhil** bearing **H.T.No: 1608-20-733-084**, **Aerpula Arundhathi** bearing **H.T.No: 1608-20-733-109** ,hereby certify that the major project report entitled **"An Ensemble Model For Glioma Sub Region Segmentation With CNN's"** is submitted in the partial fulfilment of the required for the award of the degree of Bachelor of Engineering in Computer Science and Engineering.

This is a Record of bonafide work carried out by us under the guidance of Mrs K Bhagya Laxmi, Assistance Professor, Matrusri Engineering College, Saidabad. The results embodied in this report have not been reproduced/copied from any source. The results embodied in this report have not been submitted to any other university or institute for the award of any other degree or diploma.

| | |
|---|---|
| A. Yugendhar Reddy | (1608-20-733-062) |
| P. Akhil | (1608-20-733-084) |
| A. Arundhathi | (1608-20-733-109) |

Dept. of CSE,
MECS

# Index Page

## CONTENTS

# ACKNOWLEDGEMENT

We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during our major project work. It's our pleasure to acknowledge the help of all those individuals.

We would like to thank our project guide**, Mrs. K. Bhagya Laxmi,** Assistant Professor for his guidance and help throughout the development of this project work by providing us with required information and support. Without his guidance, cooperation and encouragement, we couldn't have learnt many new things during our major project tenure.

Firstly, we would like to thank **Dr. P. Vijaya Pal Reddy,** HOD, CSE DEPT for his encouragement and valuable guidance in bringing shape to dissertation.

We express our sincere thanks to **Dr. D. Hanumanth Rao**, Principal, Matrusri Engineering College, Saidabad, Hyderabad, for his encouragement and constant help.

We extend our sincere thanks to all the teaching and non-teaching staff of Computer Science and Engineering Department for their support and encouragement.

Last but not least, We wish to acknowledge our friends and family members for giving moral strength moral strength and helping us to complete this dissertation.

| | |
|---|---|
| **A. Yugendhar Reddy** | **(1608-20-733-062)** |
| **P. Akhil** | **(1608-20-733-084)** |
| **A. Arundhathi** | **(1608-20-733-109)** |

**Dept. of CSE,**
**MECS**

## ABSTRACT

Glioma tumor sub-region segmentation is pivotal for precise diagnosis and treatment planning, delineating distinct regions within gliomas. Advanced computational techniques, including machine learning and deep learning algorithms, have revolutionized this process, enhancing accuracy and efficiency. By leveraging sophisticated imaging modalities like MRI and PET scans, clinicians can identify and characterize glioma sub-regions with unprecedented detail, enabling comprehensive assessment of tumor heterogeneity. Accurate segmentation facilitates ongoing monitoring of disease progression and empowers clinicians to tailor treatment strategies to individual patient needs, ultimately improving outcomes in the management of gliomas

Gliomas are the most common primary brain tumors, and accurate segmentation of glioma subregions from medical imaging data is crucial for treatment planning and monitoring. In this study, we propose an ensemble model for glioma subregion segmentation using a combination of U-Net, InceptionResNetv2, and WNet architectures.

# LIST OF FIGURES

# 1.INTRODUCTION

## 1.1 Introduction to the Project Work

In this project, we propose to develop an ensemble model for glioma subregion segmentation by integrating the strengths of U-Net, WNet, and InceptionResNetv2 architectures. The ensemble model aims to combine the complementary features of these architectures to improve segmentation accuracy, robustness, and generalization performance. By leveraging the diversity of predictions from multiple models, the ensemble model seeks to overcome limitations associated with individual models and produce more reliable segmentation results.

In recent years, deep learning-based segmentation methods have emerged as promising tools for automating the segmentation of glioma subregions from MRI data. Among these methods, U-Net, WNet, and InceptionResNetv2 have demonstrated significant success in medical image segmentation tasks. U-Net leverages skip connections to preserve spatial information, WNet employs an efficient encoderdecoder architecture with symmetric skip connections, and InceptionResNetv2 utilizes residual connections to facilitate deeper network architectures.

## 1.2 Glioma

Gliomas are tumors that arise from the glial cells, which are the supportive cells of the brain and spinal cord. They can occur anywhere in the central nervous system (CNS) and are categorized based on their histological features, molecular characteristics, and clinical behavior. Gliomas are the most common primary brain tumors in adults and are associated with significant morbidity and mortality. Glioma is one of the lethal brain malignancies, and it may severely damage the nervous system and endanger patients. Early diagnoses, treatments and interventions of glioma are hot research topics due to the high incidence of glioma. Surgery is the primary treatment method for glioma, and clear boundary detection is the prerequisite for a successful glioma surgery. Besides, tumor volume analysis is also very essential for evaluating the progression of the disease.

## 1.3 Sub Regions of Glioma

Subregion segmentation of gliomas aims to identify and delineate specific components within the tumor mass, each with distinct biological and clinical significance. The main subregions of gliomas typically include:

**Enhancing Tumor (ET):** This subregion represents the actively proliferating portion of the tumor and is characterized by contrast enhancement on T1weighted MRI images following gadolinium administration. The enhancing tumor region corresponds to areas of disrupted blood-brain barrier and increased vascularity.

**Non-Enhancing Tumor (NET):** Also known as the non-enhancing or infiltrative tumor, this subregion comprises tumor cells that infiltrate surrounding brain tissue without causing disruption of the blood-brain barrier. Non-enhancing tumor regions are typically hypointense or isointense on T1-weighted MRI images and hyperintense on T2-weighted or FLAIR (Fluid-Attenuated Inversion Recovery) images.

**Necrotic Core (NC):** The necrotic core of a glioma consists of areas of tissue necrosis and cell death, often resulting from inadequate blood supply and hypoxia within the tumor mass. Necrotic regions typically appear hypointense on T1weighted MRI images and hyperintense on T2weighted or FLAIR images, with central regions of fluid-filled cavities or cystic spaces.

**Peritumoral Edema (ED):** Peritumoral edema refers to the vasogenic edema that surrounds the glioma mass and results from disruption of the blood-brain barrier, leading to extravasation of fluid and proteins into the surrounding brain tissue. Edematous regions appear hyperintense on T2weighted and FLAIR images and are characterized by increased signal intensity and mass effect.

## 1.4 Importance of Glioma Sub Region Segmentation

•**Treatment Planning:** Accurate delineation of glioma subregions is essential for guiding treatment planning strategies. Different subregions may require varying treatment modalities, such as surgery, radiation therapy, chemotherapy, or targeted therapies

•**Surgical Navigation:** Intraoperative navigation systems utilize preoperative imaging data, including glioma subregion segmentation maps, to assist neurosurgeons in accurately localizing tumor boundaries during surgical resection.

•**Prognostic Stratification:** Glioma subregion segmentation provides valuable information for prognostic stratification and risk assessment in patients with gliomas. Quantitative analysis of subregion volumes, spatial distribution, and imaging biomarkers helps clinicians predict disease progression, overall survival, and treatment response, guiding patient counseling and follow-up care.

•**Monitoring Treatment Response:** Serial assessment of glioma subregion segmentation over the course of treatment enables clinicians to monitor treatment response, assess therapeutic efficacy, and detect disease progression or recurrence.

•**Research and Clinical Trials:** Glioma subregion segmentation serves as a critical endpoint in research studies and clinical trials evaluating novel treatment modalities, therapeutic agents, and imaging biomarkers. By standardizing segmentation protocols and leveraging advanced imaging technologies, researchers can advance our understanding of glioma biology, improve patient outcomes, and accelerate the development of innovative therapies. glioma subregion segmentation plays a pivotal role in personalized medicine, precision oncology, and translational research efforts aimed at improving the diagnosis, treatment, and management of patients with gliomas. By harnessing the power of advanced imaging techniques and computational algorithms, clinicians and researchers can optimize patient care pathways, enhance treatment outcomes, and ultimately improve the quality of life for individuals affected by glioma.

## 1.5 Modules

•Data Collection and Preprocessing

•Model Architecture Design

•Training

•Evaluation

•Deployment

## Module Description:-

• Data Collection and Preprocessing: Collecting multi-modal MRI datasets containingglioma images with annotated subregions Preprocessing the raw MRI data to remove noise,artifacts, and inconsistencies, ensuring data quality and integrity.

• Model Architecture Design: Designing the ensemble model architecture that combines U-Net, WNet, and InceptionResNetv2 architectures to leverage their complementary strengths in glioma subregion segmentation. Determining how the individual models will be integrated into the ensemble framework, including fusion technique weighted averaging.

• Evaluation: Visualizing segmented subregions overlaid on original MRI images and comparing them against ground truth annotations to assess segmentation accuracy and consistency. Conducting cross-validation experiments to assess the

generalization performance of the ensemble model across different patient cohorts and imaging protocols.

• Training: Training individual U-Net, WNet, and InceptionResNetv2 models on the preprocessed MRI dataset using appropriate optimization algorithms and loss functions. Integrating the trained models into the ensemble framework and finetuning ensemble parameters, such as fusion weights or decision thresholds, to optimize segmentation performance.

• Deployment: Deploying the trained ensemble model as a part of a software tool or pipeline for automated glioma subregion segmentation in clinical practice. Conducting clinical validation studies in collaboration with domain experts to evaluate the real-world performance.

# 2.LITERATURE SURVEY

## Relevant Studies:

**"Brain cancer in the world: an epidemiological review" (2019) by K. K. Farmanfarma:**

This study provides a comprehensive overview of the global epidemiology of brain cancer. The study covers various aspects such as incidence rates, mortality rates, geographical distribution, and trends over time. By synthesizing data from multiple sources, the authors offer valuable insights into the burden of brain cancer worldwide, highlighting disparities across regions and demographic groups. However, a limitation of this review is the potential for bias or inconsistency in the data sources used, which could affect the accuracy and reliability of the findings.

**"Brain tumor detection using shape features and machine learning algorithms" by D.N.George :**

This study presents a thorough investigation into the application of shape features and machine learning algorithms for the detection of brain tumors. The study explores the utilization of various shape features extracted from medical imaging data, coupled with machine learning techniques, to develop an automated system for brain tumor detection. By examining the effectiveness of different algorithms, the authors contribute valuable insights into the potential of this approach for accurate and efficient diagnosis of brain tumors. However, a limitation of this survey is the lack of extensive validation and testing on diverse datasets. The study primarily focuses on evaluating the proposed methods using a specific dataset, potentially limiting the generalizability of the findings to broader populations or imaging modalities. Future research should address this limitation by conducting comprehensive validation studies across multiple datasets to assess the robustness and applicability of the proposed approach in real-world clinical settings.

**"The 2016 World Health Organization classification of tumors of the central nervous system: a summary" (2016) by D.N.Louis:**

This study provides a comprehensive summary of the updated classification system for central nervous system (CNS) tumors introduced by the World Health Organization (WHO) in 2016. The study offers valuable insights into the revised diagnostic criteria, molecular markers, and classification schemes for various types of CNS tumors, facilitating improved clinical diagnosis and management. However, a limitation of this review is the potential for variations in the interpretation and implementation of the WHO classification criteria across different healthcare settings and regions. Additionally, the summary may not cover all nuances and complexities associated with each tumor type, and further detailed analysis and validation studies may be required to fully understand the implications of the updated classification system in clinical practice. Therefore, future research should aim to address these limitations by conducting comprehensive evaluations and validations of the WHO classification system in diverse patient populations and healthcare settings to ensure its effectiveness and reliability in guiding clinical decision-making for CNS tumors.

**"Automatic brain tumor detection and segmentation using U-Net based**

**fully convolutional networks" (2019) by H.Dong:**

This review article presents a thorough examination of the application of deep learning techniques, particularly U-Net based fully convolutional networks, for the automatic detection and segmentation of brain tumors. The study investigates the effectiveness of these methods in accurately identifying and delineating tumor regions from medical imaging data, offering potential benefits for clinical diagnosis and treatment planning. However, a limitation of this research lies in the generalizability of the proposed approach across different datasets and imaging modalities. Since the study primarily focuses on a specific dataset or datasets with similar characteristics, the performance of the proposed method may not be robust when applied to diverse datasets with varying image qualities, acquisition parameters, or tumor types. Therefore, future research should aim to address this limitation by conducting validation studies on more heterogeneous datasets to assess the robustness and scalability of the proposed approach in real-world clinical settings.

**"Brain tumor segmentation and overall surviva period prediction in glioblastoma multiforme using radiomic features" (2019) by S.Bose:**

This review article presents a thorough investigation into the application of radiomic features for brain tumor segmentation and overall survival period prediction in glioblastoma multiforme (GBM) patients. By utilizing advanced computational techniques, the authors demonstrate the potential of radiomic features extracted from medical images for improving the accuracy of tumor segmentation and predicting patient survival outcomes. This research contributes to the ongoing efforts to leverage machine learning and image processing methods for enhancing diagnosis and prognostication in GBM. However, a limitation of this study lies in the generalizability of the findings, as the analysis is based on a specific dataset or cohort of GBM patients. The effectiveness of the proposed approach may vary when applied to different patient populations or imaging protocols, highlighting the need for further validation and replication across diverse datasets to ensure its robustness and reliability in clinical practice. Additionally, while radiomic features offer valuable insights into tumor characteristics, their interpretation and clinical translation may require integration with other clinical and molecular data to enhance their utility for personalized medicine approaches in GBM management.Top of Form

**"Brain Tumor Segmentation from MRI Images Using Deep Learning Framework" (2019) by S.Das:**

The study explores various deep learning architectures, methodologies, and algorithms employed in brain tumor segmentation tasks, highlighting their strengths and limitations. Das discusses the challenges faced in accurately segmenting brain tumors, such as dealing with heterogeneous tumor appearances, small tumor sizes, and the presence of noise and artifacts in MRI images. However, a limitation of this literature survey is the potential lack of depth in the discussion of certain advanced or emerging deep learning techniques and their specific applicability to brain tumor segmentation. Therefore, future research should aim to address these limitations by providing more indepth analyses and considering the latest advancements in deep learning for brain tumor segmentation.

# 3.ANALYSIS

## 3.1 Drawbacks of Existing Approaches:

Several existing systems and approaches for glioma subregion segmentation have been developed, utilizing various techniques ranging from traditional image processing methods to state-of-the-art deep learning architectures. While existing systems for glioma subregion segmentation have shown promising results, they also exhibit certain limitations that warrant consideration. Some of these limitations include:

### 3.1.1. U-Net-Based Approaches:

High Computational Cost: Training U-Net models, especially in 3D, can be computationally intensive and require significant computational resources, limiting their scalability and practical deployment in clinical settings.

### 3.1.2. DeepMedic:

Complexity and Interpretability: DeepMedic combines deep convolutional neural networks with conditional random fields, resulting in a complex model architecture that may be challenging to interpret and debug. Understanding the factors influencing segmentation outcomes and model decisions can be difficult due to the model's high dimensionality and nonlinear behavior.

### 3.1.3. Attention Mechanism-Based Models:

Computational Overhead: Models incorporating attention mechanisms, such as Attention U- Net or Dense Attention U-Net, may introduce additional computational overhead due to the need to compute attention maps or attend to relevant image regions. This increased computational complexity can impact inference speed and model deployment in resource- constrained environments.

### 3.1.4. Graph-Based Methods:

Computational Complexity: Graph-based methods, such as graph cuts or graph convolutional networks (GCNs), may involve computationally intensive optimization or inference procedures, particularly for large-scale graphs or high-dimensional data. Managing computational complexity and scalability is essential for practical deployment of graph-based segmentation models.

## 3.2 Proposed System:

Our Proposed Method proposes a hybrid deep learning model Convolutional Neural Network by combing three CNN models ( InceptionResNetv2, UNet, WNet ) for classifying and predicting Glioma.Using ensemble technique average weighting.

The ensemble model combining CNN models such as U-Net, InceptionResNetv2, and WNet for glioma subregion segmentation offers several advantages over individual models

Our Proposed Method proposes a hybrid deep learning model Convolutional Neural Network by combing three CNN models ( InceptionResNetv2, UNet, WNet ) for classifying and predicting Glioma.Using ensemble technique average weighting.

The ensemble model combining CNN models such as U-Net, InceptionResNetv2, and WNet for glioma subregion segmentation offers several advantages over individual models:

- Improved Segmentation Accuracy: Ensemble models can leverage the strengths of multiple CNN architectures to achieve higher segmentation accuracy. By combining the predictions of U-Net, InceptionResNetv2, and WNet, the ensemble model can effectively mitigate errors and uncertainties inherent in individual models, leading to more reliable segmentation results.
- Enhanced Robustness: Ensemble models are inherently more robust to variations in input data and model architectures. By aggregating predictions from multiple sources, the ensemble model can reduce the impact of outliers, noise, or artifacts in the input images, resulting in more stable and consistent segmentation performance.
- Complementary Features: Each CNN model (U-Net, InceptionResNetv2, WNet) has unique architectural features and capabilities. U-Net is known for its ability to preserve spatial details through skip connections, InceptionResNetv2 facilitates deeper network architectures with residual connections, and WNet offers

- An efficient encoding-decoding architecture with symmetric skip connections. By combining these complementary features, the ensemble model can capture a more comprehensive representation of glioma subregions, improving segmentation accuracy.
- Mitigation of Model Biases: Individual CNN models may exhibit biases or limitations in capturing certain features or subregions within gliomas. By ensembling

multiple models, the ensemble model can mitigate these biases and achieve a more balanced representation of different subregions, leading to more accurate and comprehensive segmentation outcomes.

- Robustness to Model Variability: Ensemble models are less sensitive to variability in individual model performance or hyperparameter settings. Even if one CNN model performs poorly on certain subregions or cases, the ensemble's collective decision-making process can compensate for such variations and produce more consistent results.
- Improved Generalization: Ensemble models often exhibit superior generalization performance compared to individual models. By combining predictions from diverse CNN architectures trained on different subsets of data, the ensemble model can better generalize to unseen datasets, patient cohorts, and imaging protocols, enhancing its applicability in real-world clinical settings.
- Reduced Overfitting: Ensemble models can help mitigate overfitting by reducing the variance in predictions across multiple models. By combining predictions from different CNN architectures, the ensemble model can achieve a more robust and generalized representation of glioma subregions, minimizing the risk of overfitting to specific training data.
- Overall, These advantages make ensemble models a promising approach for tomating glioma subregion segmentation and facilitating clinical decision-making in neuro-oncology.

## 3.3 Objectives of the Project :

- Preprocessing the Dataset for feature extraction and classification by applying a median filter, and normalization to reduce noise, and improve performance.
- Utilising the features of DataGerators in keras which resizes them,  normalizes the input data and also provide different variations of input Dataset
- For segmentation U-Net ,  InceptioResNetV2 ,  WNet method is used combinely using weighting average technique as an Ensembling Model .
- Compare metrics such as accuracy, sensitivity, specificity, Recall , precision and Mean IOU to evaluate the performance

# 4.DESIGN

**UML Diagrams:**

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language for:

•Visualizing

•Specifying

•Constructing

•Documenting the artifacts of a software intensive system.

The UML is a language that provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system.

There are two broad categories of diagrams and they are again divided into structural diagrams and behavioral diagrams. The structural diagrams represent the static aspect of the system. The four structural diagrams are class diagram, object diagram, component diagram, deployment diagram. Behavioral diagrams basically capture the dynamic aspect of a system. Types of behavioral diagrams are use case diagrams, sequence diagrams, collaboration diagrams, state chart diagrams, and activity diagram. Some of the frequently used use case diagrams in software development are as below:

## 4.1 Use case:

A use case is defined as a set of sequences of actions performed by an actor to achieve a specific result. A use case contains an actor. An actor refers to various people that use the system.

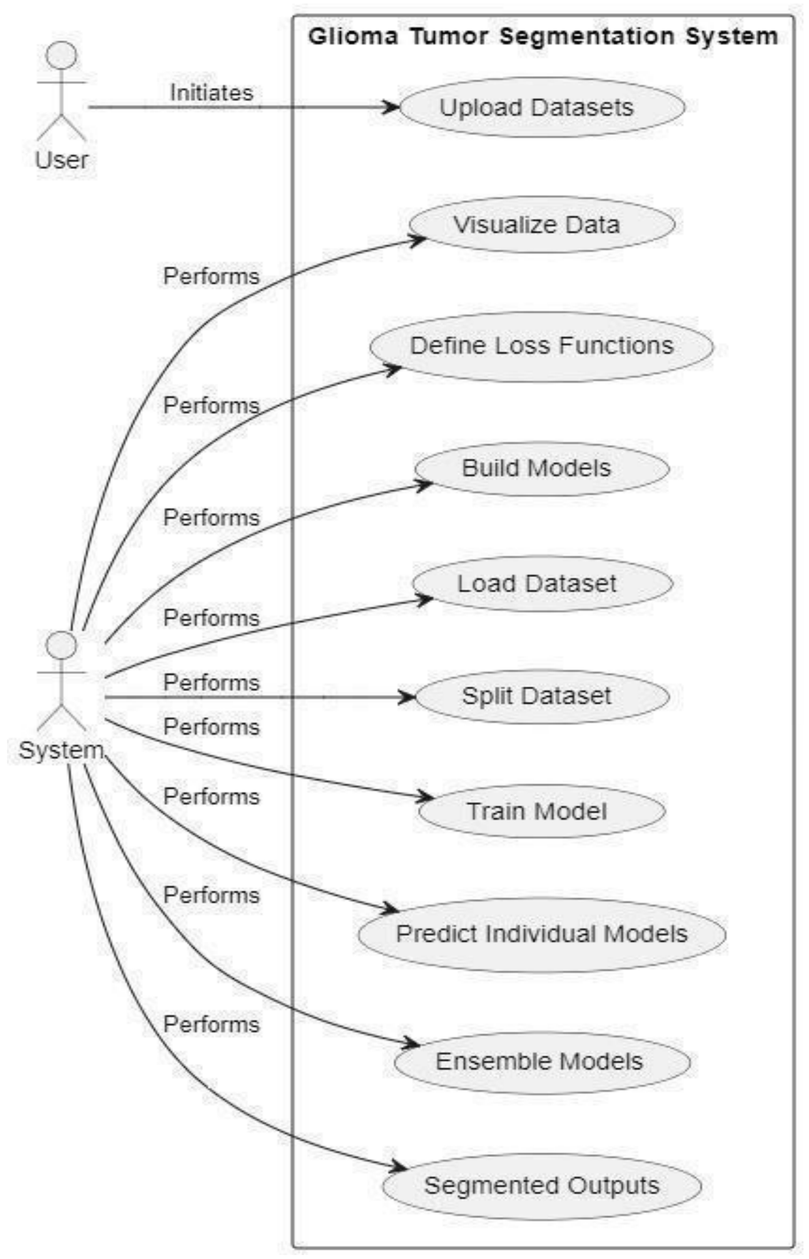


Fig 4.1 Usecase diagram

## 4.2 Class diagram:

In software engineering, a class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. It represents the functionality of various requirements in an applications

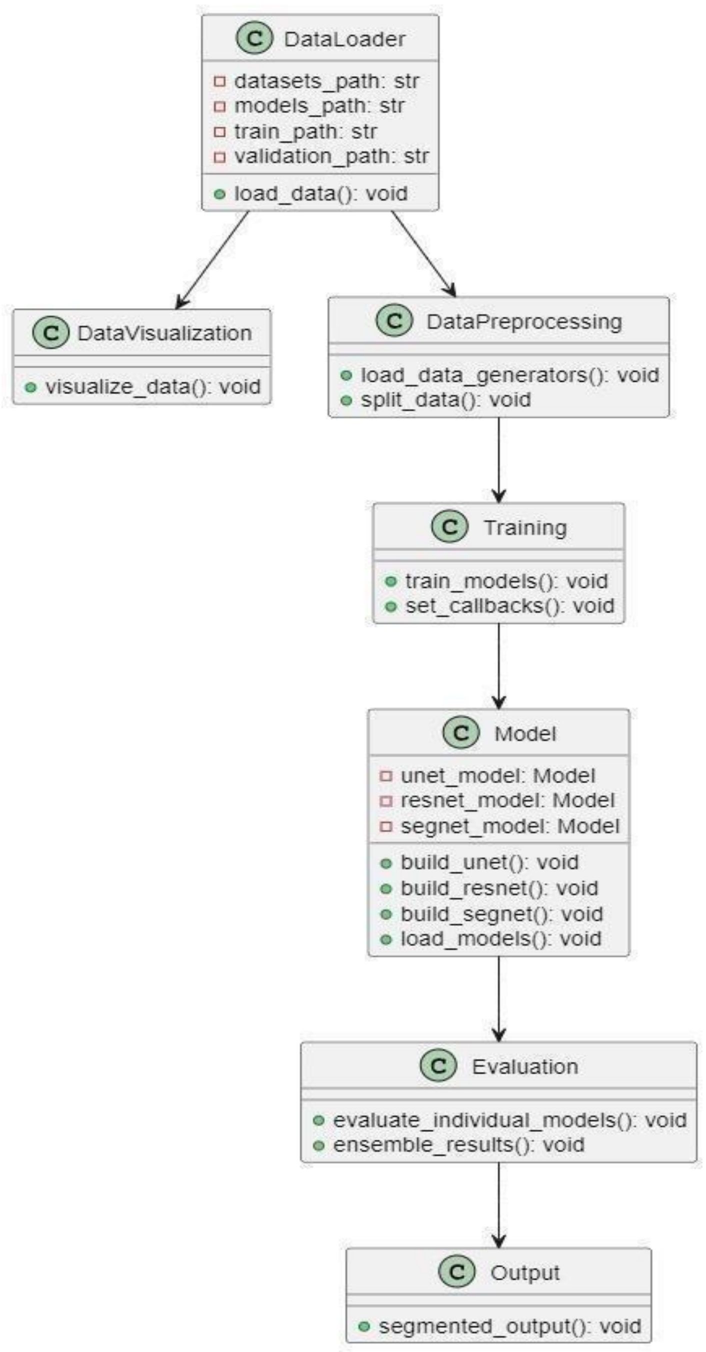


Fig 4.2 class diagram

## 4.3 State Machine diagram:

A state machine diagram in UML (Unified Modeling Language) is used to model the behavior of a system or an object in terms of states, transitions between those states, and events triggering those transitions.

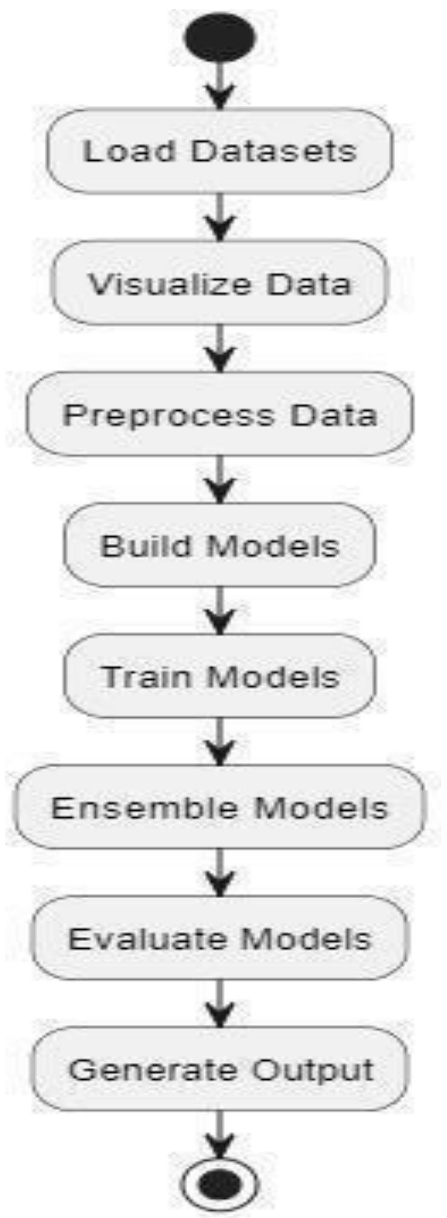


Fig 4.3 State Machine diagram

## 4.4 Sequence diagram:

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.

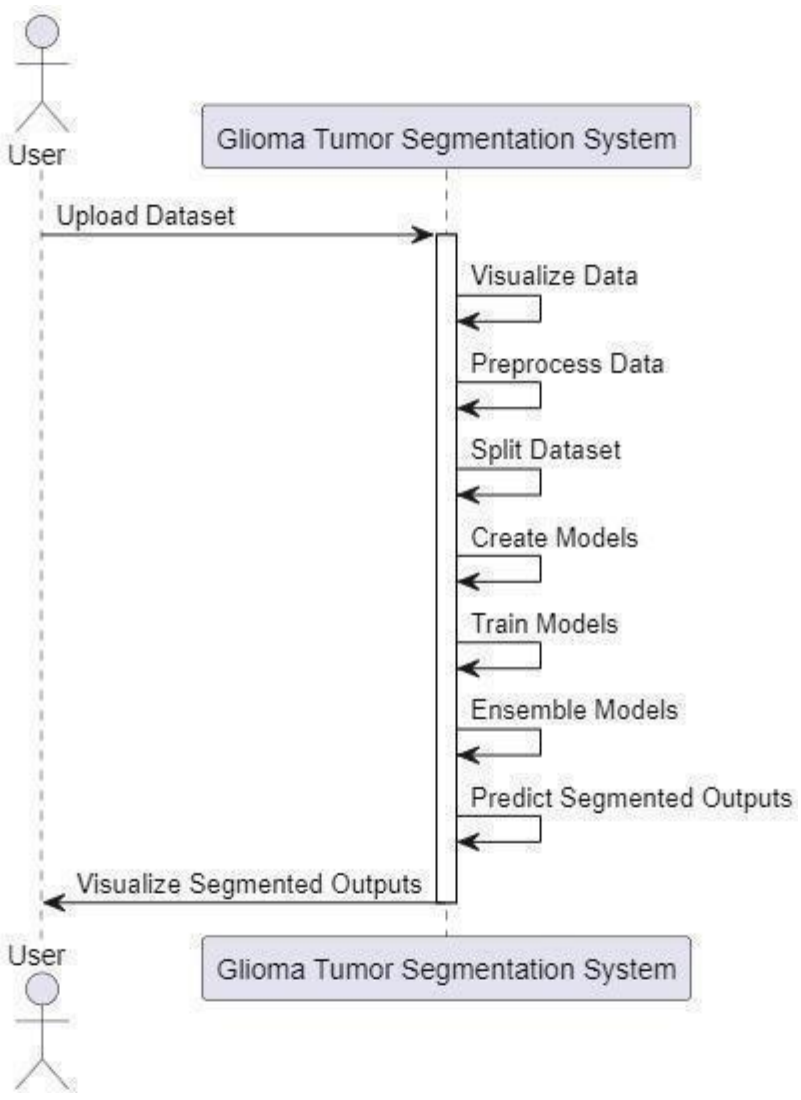

User
Glioma Tumor Segmentation System
Upload Dataset
Visualize Data
Preprocess Data
Split Dataset
Create Models
Train Models
Ensemble Models
Predict Segmented Outputs
Visualize Segmented Outputs
User
Glioma Tumor Segmentation System

Fig 4.4 Sequence diagram

# 5.IMPLEMENTATION

The implementation of a project focused on developing an ensemble model for glioma subregion segmentation using U-Net, InceptionResNetv2, and WNet involves several steps. Here's an outline of the implementation process:

## 5.1 Uploading Datasets and Models:

The trained segmentation models, including U-Net, InceptionResNetv2, and WNet architectures, are uploaded from GitHub repositories. Researchers access the repositories, clone or download the models to their local machines, and extract the model. Platforms like GitHub, GitLab, or cloud storage services are chosen for hosting the models. A new repository or branch is created specifically for the models, and the model files are uploaded to the repository or branch.

The datasets utilized in the project are sourced from the BraTS (Brain Tumor Segmentation) challenge, a widely recognized
benchmark dataset in the field of medical image analysis. the dataset is
organized into a structured format, separating MRI scans and corresponding segmentation labels for glioma subregions. To upload the dataset, a secure and reliable platform is chosen for hosting, such as cloud storage services or dedicated data repositories. Platforms like Google Drive, Dropbox, Kaggle Datasets, or Zenodo are commonly utilized for this purpose. The dataset files are uploaded to the selected hosting platform, ensuring proper organization and documentation. A README file accompanies the dataset, providing detailed information about the dataset contents, imaging modalities, patient demographics, annotation guidelines, and any preprocessing steps applied.

## 5.2 Environment Setup:

Set up the development environment with necessary software libraries, frameworks (e.g., TensorFlow, PyTorch), and hardware resources
(GPU/CPU).

- **Python Environment:**

Install Python, preferably the latest version available. Set up a virtual environment using tools like virtualenv or conda to manage dependencies and avoid conflicts between packages

.

- **Deep Learning Frameworks:**

Install deep learning frameworks such as TensorFlow or PyTorch, which provide high- level APIs for building and training neural
networks. Ensure that the chosen framework supports GPU
acceleration for faster training, especially for larger datasets and complex models.

- **GPU Support:**

If available, install GPU drivers and libraries to enable GPU acceleration. Configure the deep learning framework to utilize the GPU for accelerated training. This may involve setting environment variables or modifying configuration files.

- **Data Processing Libraries:**

Install libraries for data processing and manipulation, such as NumPy, Pandas (for medical image processing).

**Image data descriptions**

All BraTS multimodal scans are available as NIfTI files (.nii.gz) -> commonly used medical imaging format to store brain imagin data obtained using MRI and describe different MRI settings

1. T1 : T1-weighted, native image, sagittal or axial 2D acquisitions, with 1–6 mm slice thickness.
2. T1c : T1-weighted, contrast-enhanced (Gadolinium) image, with 3D acquisition and 1 mm isotropic voxel size for most patients.
3. T2 : T2-weighted image, axial 2D acquisition, with 2–6 mm slice thickness.
4. FLAIR : T2-weighted FLAIR image, axial, coronal, or sagittal 2D acquisitions, 2–6 mm slice thickness.

Data were acquired with different clinical protocols and various scanners from multiple (n=19) institutions.

## 5.3 Data Preprocessing:

### 5.3.1 Data Visualization:

Visualization techniques are employed to provide insights into the spatial distribution of intensity values, structural variations, and pathological features present in the images. By visualizing MRI scans from multiple modalities, researchers gain a comprehensive understanding of glioma heterogeneity and subregion boundaries. Through interactive visualization tools and techniques, such as matplotlib, seaborn, or medical image visualization libraries, intricate details within the images can be explored and analyzed. This enables researchers to identify potential biomarkers, delineate tumor boundaries, and assess the efficacy of segmentation algorithms. Data visualization not only enhances the interpretability of MRI data but also aids in the development and evaluation of segmentation models for accurate glioma subregion delineation.

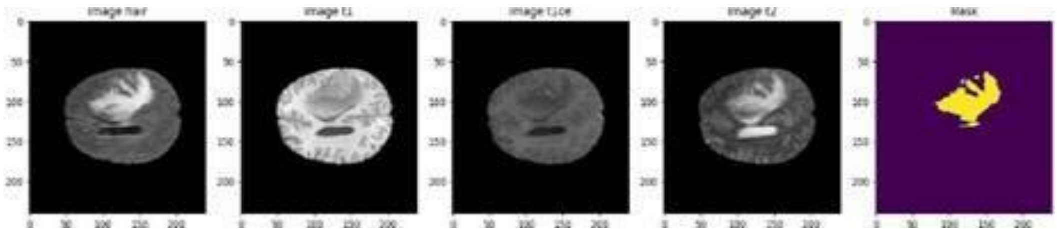


Fig.5.3.1 Data visualization

### 5.3.2 Showing whole nifti data -> printing each slice from 3d data:

Converting whole NIfTI (Neuroimaging Informatics Technology Initiative) pictures to slices is a crucial preprocessing step essential for subsequent analysis and modeling. NIfTI images typically contain volumetric data representing MRI scans, including multiple slices across different imaging modalities. By converting the entire volumetric NIfTI images into individual slices, researchers can effectively analyze and process each slice independently, enabling finer-grained analysis and segmentation of glioma subregions. This conversion process involves extracting 2D slices along specific axes from the 3D volumetric data. Each resulting slice represents a 2D image containing spatial information from the corresponding 3D volume. By generating a comprehensive set of slices from the NIfTI images across all modalities (T1, T1c, T2, FLAIR), researchers can facilitate feature extraction, visualization, and model training for accurate glioma subregion segmentation. Additionally, the conversion to slices enables the application of convolutional neural networks (CNNs) and other deep learning architectures that are commonly used for medical image analysis tasks.

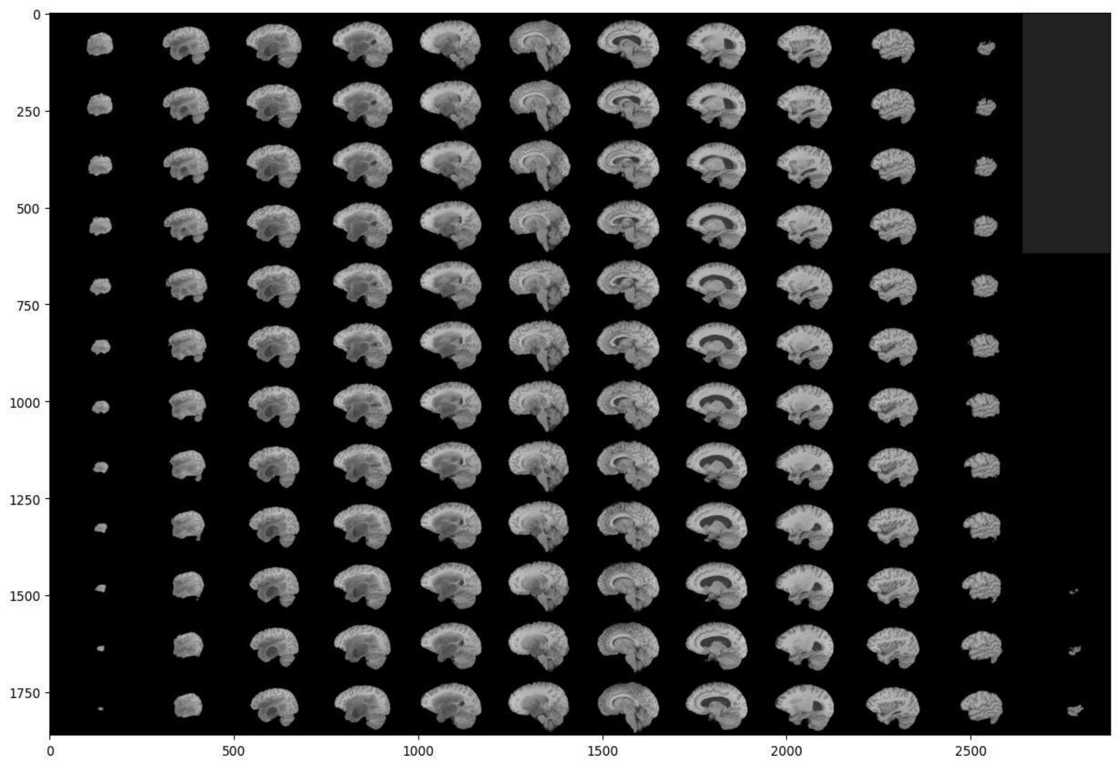


Fig 5.3.2 Showing 2D Slices

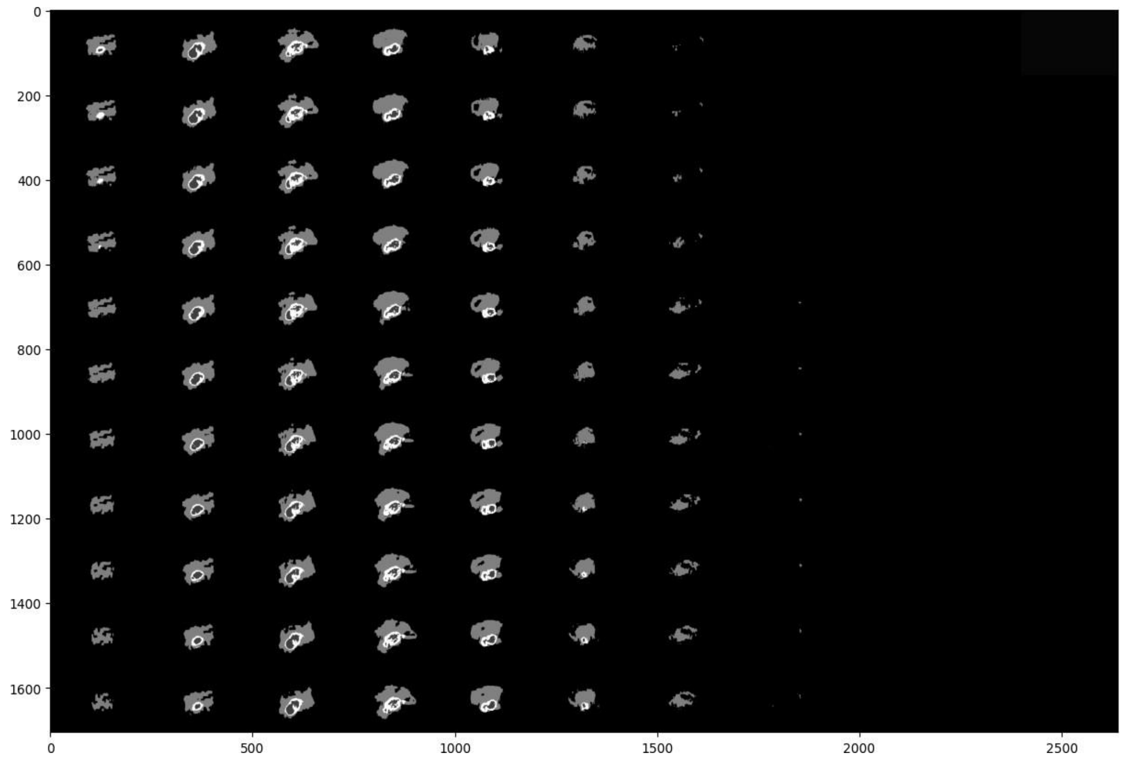


Fig 5.3.3 Showing segments of tumour region in each slice

## 5.3 Loss Functions:

The choice and design of the loss function play a pivotal role in guiding the training process of deep learning models, such as U-Net, InceptionResNetv2, and WNet. The loss function serves as a measure of dissimilarity between the predicted segmentation maps and the ground truth labels, guiding the optimization process towards minimizing segmentation errors and improving accuracy.

Typically, a multi-class segmentation problem like glioma subregion segmentation requires the use of a loss function that can handle pixel-wise classification across multiple classes. The commonly used loss functions include cross-entropy loss, Dice loss, and their variants tailored for segmentation tasks. Cross-entropy loss measures the discrepancy between the predicted probability distributions and the true labels, penalizing misclassifications. On the other hand, Dice loss

quantifies the overlap between the predicted and ground truth segmentation masks, focusing on the spatial agreement between the two.

In this project, a hybrid loss function may be employed to leverage the strengths of both cross-entropy and Dice loss. The hybrid loss combines the pixel-wise classification accuracy captured by crossentropy loss with the spatial similarity measured by Dice loss, providing a comprehensive optimization objective for training the segmentation models. Additionally, class-weighted variants of the loss function may be utilized to address class imbalance issues commonly encountered in medical imaging datasets.

During the training phase, the loss function is computed for each pixel in the predicted segmentation maps, comparing them against the corresponding ground truth labels. The gradients of the loss function are then backpropagated through the network, updating the model parameters to minimize the loss and improve segmentation

performance iteratively. Through this iterative optimization process, the segmentation models learn to capture relevant features and spatial patterns indicative of glioma subregions, ultimately leading to more accurate segmentation results.

## 5.4 Training and Validation:

Train the ensemble model using the preprocessed MRI dataset and evaluate its performance on a separate validation dataset.

Monitor training progress, including loss convergence, accuracy, and validation metrics, to ensure model stability and convergence.

Qualitatively assess segmentation results through visual inspection of segmented subregions overlaid on original MRI images and comparison against ground truth annotations.

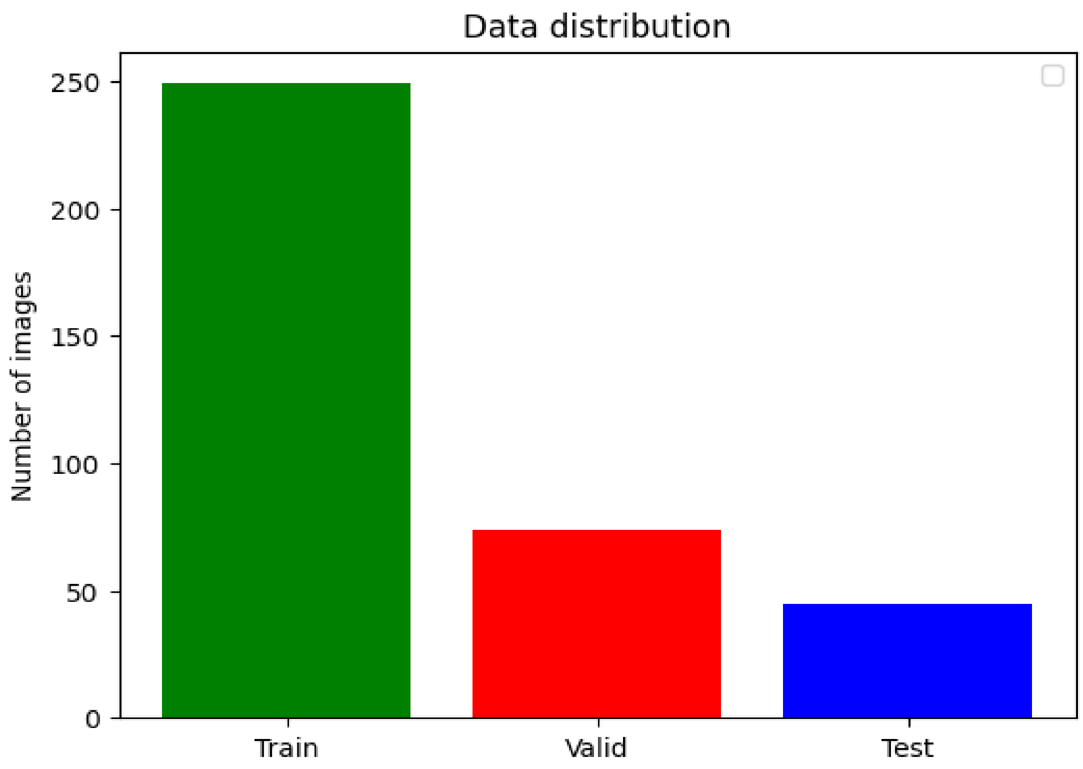


Fig 5.4 Division of Datasets into Train , Validation and testing Datapoints

## 5.5 Model Development:

Design and implement the ensemble model architecture that combines U- Net, InceptionResNetv2, and WNet architectures for glioma subregion segmentation. Train individual UNet, InceptionResNetv2, and WNet models on the preprocessed MRI dataset using appropriate optimization algorithms and loss functions. Finetune hyperparameters for each model architecture, including network depth, learning rate, batch size, and regularization parameters.

### 5.5.1 UNet:

U-Net is a convolutional neural network (CNN) architecture designed for biomedical image segmentation, particularly for tasks where highresolution segmentation masks are required, such as in medical image analysis. It was introduced by Olaf Ronneberger, Philipp Fischer, and Thomas Brox The model begins with an input layer that accepts images of shape (IMG_SIZE, IMG_SIZE, 2), where IMG_SIZE is the spatial dimension and 2 is the number of input channels.

**Downsampling Path (Encoder)**

The encoder path consists of repeated application of two convolutional layers, each followed by a ReLU activation function and he_normal kernel initializer, and a max-pooling layer to downsample the feature maps.

**First Block:** Applies two convolutional layers with 32 filters and then a max-pooling layer.

**Second Block:** Consists of two convolutional layers with 64 filters, followed by a max-pooling layer.

**Third Block:** Includes two convolutional layers with 128 filters, followed by max-pooling.

**Fourth Block**: Contains two convolutional layers with 256 filters, followed by max-pooling.

**Fifth Block:** Uses two convolutional layers with 512 filters, followed by a dropout layer with a dropout rate of 0.2.

**Bottleneck:** The bottleneck layer serves as a bridge between the encoder and decoder. It includes two convolutional layers with 512 filters each, maintaining the highest level of feature abstraction.

**Upsampling Path (Decoder)**

The decoder path mirrors the encoder but replaces pooling with upsampling layers to progressively restore the spatial dimensions of the feature maps. Each upsampling step is followed by a concatenation with corresponding feature maps from the encoder path to ensure precise localization.

**First Up Block:** Upsamples the feature map, followed by a convolutional layer with 256 filters, then concatenates with the corresponding encoder feature map. This is followed by two convolutional layers with 256 filters.

**Second Up Block:** Upsamples and applies a convolutional layer with 128 filters, concatenates with encoder features, and includes two convolutional layers with 128 filters.

**Third Up Block:** Similar to previous blocks, but with 64 filters in convolutional layers.

**Fourth Up Block:** Again, upsampling followed by a convolutional layer with 32 filters, concatenation, and two convolutional layers with 32 filters.

**Output Layer**

Finally, the model ends with a convolutional layer with 4 filters (assuming a segmentation task with four classes) and a 1x1 kernel size, followed by a softmax activation function to produce the final segmentation map.

**Model Compilation**

The model is compiled using the categorical cross-entropy loss function, Adam optimizer with a learning rate of 0.001, and several performance metrics including accuracy, Mean Intersection over Union (IoU), dice coefficient, precision, sensitivity, specificity, and class-specific dice coefficients.

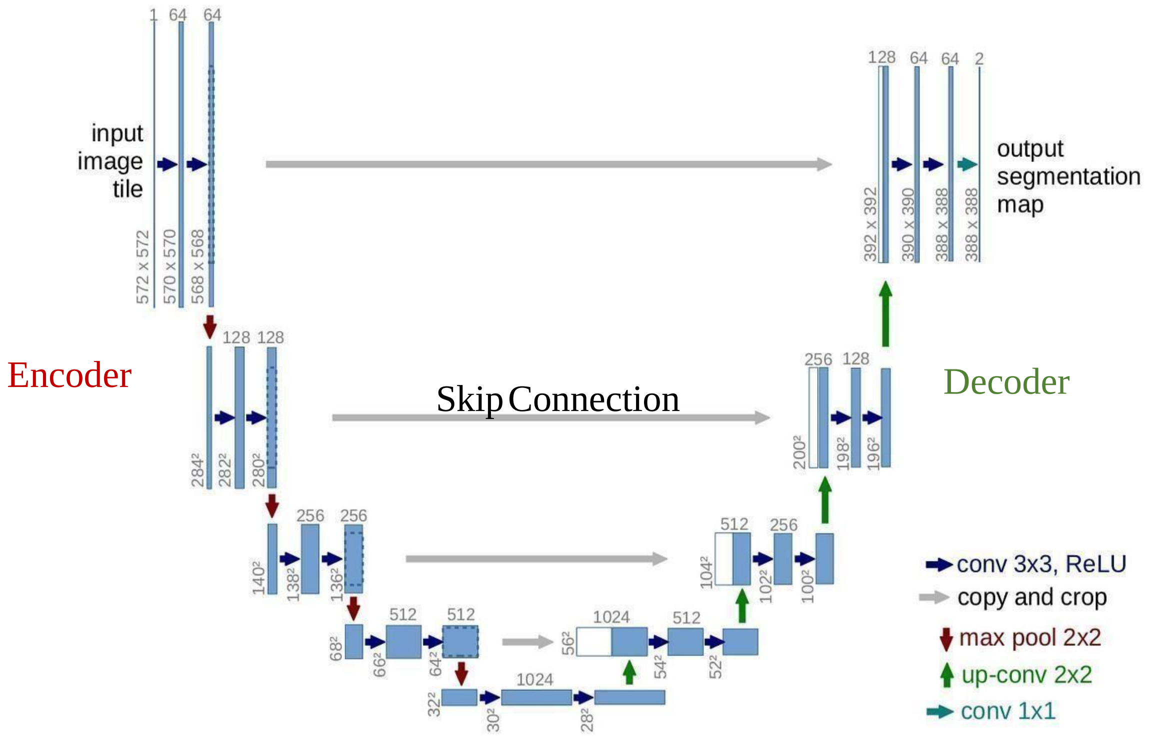


Fig 5.5.1 UNet Architecture Diagram

## 5.5.2 InceptionResNetv2:

InceptionResNetV2 is a state of the art convolutional neural network architecture that combines the Inception and ResNet modules, offering superior performance in various computer vision tasks, including image classification and object detection. This architecture introduces residual connections from the ResNet architecture into the Inception architecture, enhancing gradient flow and enabling deeper networks without suffering from vanishing gradients.

**Inception Module:** The Inception module is characterized by its parallel paths of convolutions with different kernel sizes. These paths capture features at multiple scales and resolutions simultaneously, allowing the network to efficiently learn hierarchical representations.

**ResNet Module:** The ResNet module introduces residual connections that skip one or more layers. These connections facilitate the flow of gradients during training, enabling the network to effectively learn from deeper layers without encountering degradation in performance.

**InceptionResNetV2 Architecture:**

**Stem:** The network begins with a stem module that processes the input image and extracts basic features.

**Inception Blocks:** These blocks consist of a combination of Inception modules and ResNet modules. They capture complex features at different scales while maintaining gradient flow through residual connections.

**Reduction Blocks:** These blocks reduce spatial dimensions using techniques like max pooling and convolution, helping to condense feature maps and increase computational efficiency.

**Classification Head:** The final part of the network consists of a global average pooling layer followed by fully connected layers for classification.

**Feature Hierarchy:** The combination of Inception and ResNet modules allows the network to capture detailed features at multiple scales, creating a rich feature hierarchy.

**Gradient Flow:** Residual connections facilitate the flow of gradients during training, enabling the network to effectively learn from deeper layers without suffering from vanishing gradients.

**Efficiency:** The parallel paths in Inception modules and the skip connections in ResNet modules contribute to the efficient utilization of parameters and computational resources.

**InceptionResNetV2's symmetrical architecture:** It starts with feature extraction through the stem module, followed by multiple blocks of Inception and ResNet modules for feature refinement. The network ends with a classification head for making predictions.
This architecture's combination of Inception and ResNet modules provides a powerful

framework for various computer vision tasks, offering both high accuracy and computational efficiency.

**InceptionResNetV2**

InceptionResNetV2 is an advanced neural network architecture that merges the Inception modules with Residual connections, enhancing both efficiency and accuracy in image recognition tasks. Here’s a breakdown of its implementation:

**Input Layer**

Input: Accepts images with a specific shape, typically (height, width, channels).

**Stem Block**

Stem Block: Begins with a series of convolutional and pooling layers to downsample the input and capture initial features.

Convolutional layer: 32 filters, 3x3 kernel size, stride of 2, no padding.
Convolutional layer: 32 filters, 3x3 kernel size, no padding.
Convolutional layer: 64 filters, 3x3 kernel size, padding.
Max-pooling layer: 3x3 pool size, stride of 2.
Convolutional layer: 80 filters, 1x1 kernel size.
Convolutional layer: 192 filters, 3x3 kernel size.
Max-pooling layer: 3x3 pool size, stride of 2.

**Inception-ResNet Blocks**

Inception-ResNet-A: Comprises several parallel convolutional paths with different kernel sizes (1x1, 3x3, 5x5), concatenated and followed by a residual connection.

Reduction-A: A transition block that reduces spatial dimensions using pooling and convolutional layers.

Inception-ResNet-B: Similar to Inception-ResNet-A but with more complex convolutional structures, concatenated and followed by a residual connection.

Reduction-B: Another transition block to further reduce dimensions.

Inception-ResNet-C: More complex convolutional structures with parallel paths, concatenated and followed by a residual connection.

**Final Layers:**

Average Pooling: A global average pooling layer to reduce the spatial dimensions to 1x1.
Dropout: An optional dropout layer for regularization.
Dense Layer: A fully connected layer for final classification, usually with a softmax activation for multi-class classification tasks.

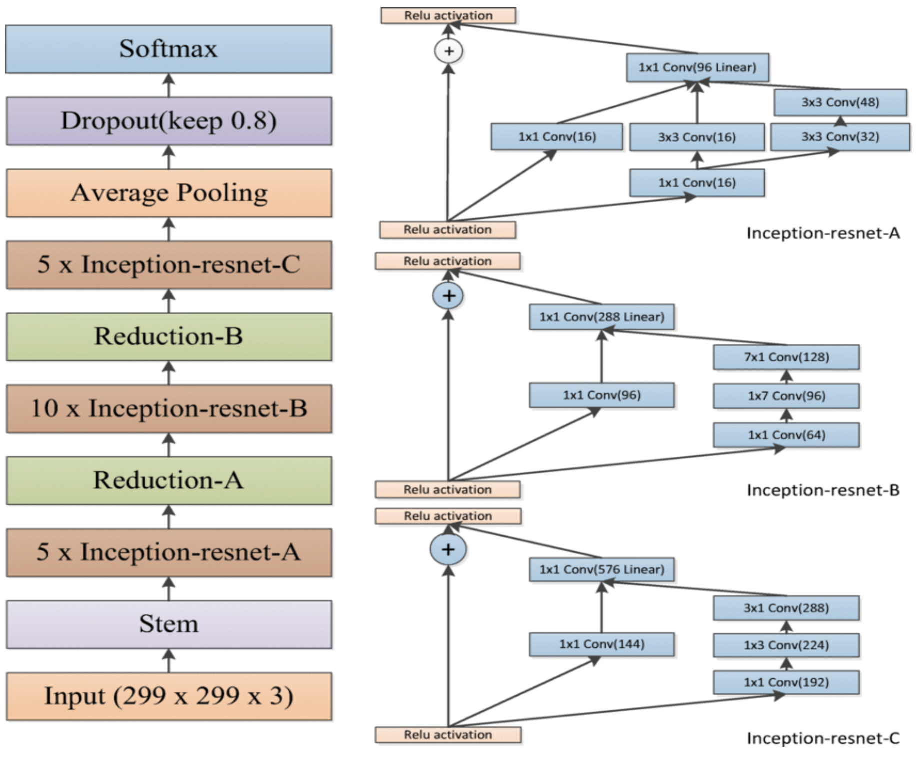


Fig 5.5.2 InceptionResNetv2 Architecture Diagram

### 5.5.3 WNet :

W-Net is a deep learning architecture designed for image segmentation tasks. It consists of two stacked U-Nets, where the output of the first U-Net is fed into the second U-Net. This setup allows for progressive refinement of the segmentation results, leveraging the strengths of two U-Net architectures to improve accuracy and detail in the segmented images.

**Working of W-Net in Glioma Subregion Segmentation**

In the context of glioma subregion segmentation, W-Net is particularly effective due to its capability to capture complex structures and fine details within MRI images. Here's how W-Net operates in this project:

W-Net is an extension of the U-Net architecture designed for unsupervised image segmentation. It consists of two U-Nets stacked together: the first U-Net performs the initial segmentation, and the second U-Net refines this segmentation.

**Input Layer**

Input: Similar to U-Net, it starts with an input layer that accepts images of shape (height, width, channels).

**First U-Net (Encoder-Decoder Path)**

**Encoder Path:** Sequential convolutional and max-pooling layers to capture features and reduce spatial dimensions.

Multiple blocks, each with two convolutional layers followed by a max-pooling layer.
Increasing number of filters (e.g., 32, 64, 128, 256) with each block.
**Bottleneck**: Convolutional layers at the deepest level to capture abstract features.

**Decoder Path:** Upsampling layers followed by convolutional layers to restore spatial dimensions and merge features from corresponding encoder layers.

Multiple blocks with upsampling, concatenation, and convolutional layers.
Decreasing number of filters (e.g., 256, 128, 64, 32).
Transition

The output of the first U-Net serves as the input to the second U-Net for refinement.

**Second U-Net (Refinement Encoder-Decoder Path)**

**Encoder Path:** Similar to the first U-Net but operates on the output from the first U-Net.
Sequential blocks of convolutional and max-pooling layers.
**Bottleneck:** Convolutional layers at the deepest level to refine features.

**Decoder Path:** Similar to the first U-Net but aims to refine the segmentation map.
Sequential upsampling and convolutional layers with concatenations.

**Final Layers**

Output Layer: A convolutional layer with filters equal to the number of segmentation classes, typically followed by a softmax activation for multi-class segmentation tasks.

**Model Compilation**

The W-Net model is compiled with appropriate loss functions for segmentation (e.g., categorical cross-entropy), optimizers (e.g., Adam), and metrics (e.g., accuracy, IoU, dice coefficient).

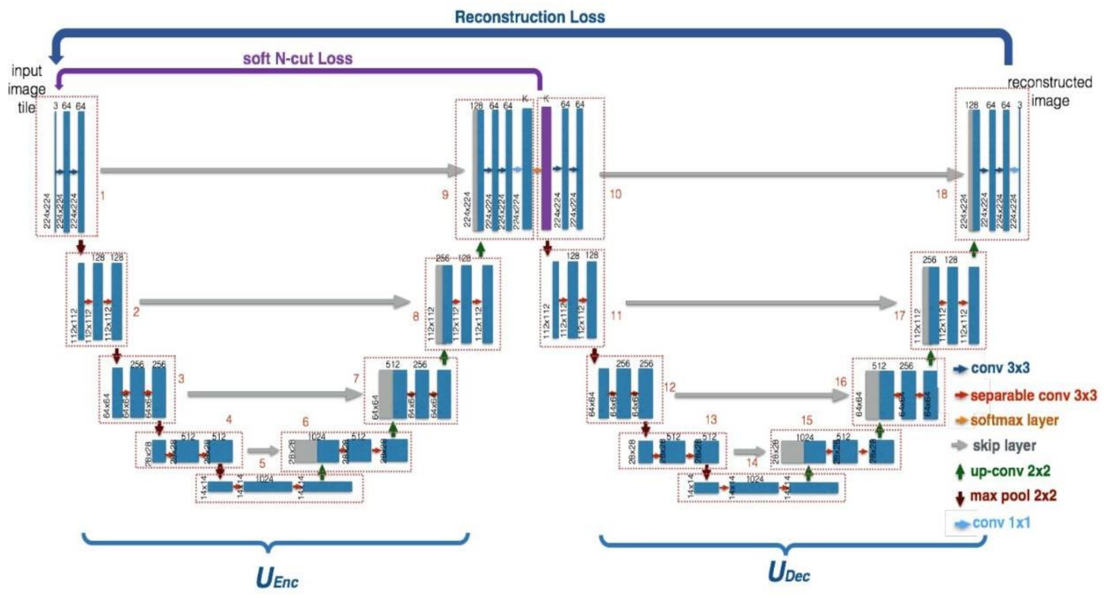


Fig 5.5.3 WNet Architecture

## 5.6 Model Import:

Model importing is a fundamental step in deploying and evaluating trained deep learning models for segmenting glioma

subregions in MRI images. After the models are trained and saved, they need to be imported into the inference environment for making predictions on new data. This process typically involves loading the model architecture and weights from storage and initializing the model object. Depending on the framework used for model training and deployment, different APIs and functions are employed for model importing.

Once imported, the segmentation models are ready for inference, where they process input MRI images and generate segmentation maps predicting the spatial distribution of glioma subregions. The imported models encapsulate the learned knowledge and patterns extracted from the training data, enabling them to generalize and make accurate predictions on unseen data. In the project, models trained using architectures such as U-Net, InceptionResNetv2, and WNet are imported, each offering unique advantages and capabilities for glioma subregion segmentation tasks.

Before model importing, it's essential to ensure compatibility between the training and inference environments, including consistent

versions of software libraries, hardware configurations, and preprocessing steps. Additionally, model importing may involve considerations for memory constraints, especially when dealing with large models or limited computational resources.

## 5.7 Evaluation

Evaluation is conducted to assess the performance and effectiveness of the imported models in accurately delineating glioma subregions in MRI images. Evaluation encompasses a series of quantitative and qualitative analyses aimed at understanding the models' strengths, limitations, and overall efficacy in clinical practice.

Quantitative evaluation involves computing various performance

metrics, such as Dice similarity coefficient, sensitivity and specificity to quantitatively measure the agreement between the predicted segmentation maps and the ground truth labels. These metrics provide insights into the models' ability to accurately capture the spatialextent and boundaries of glioma subregions, facilitating objective comparisons acrossdifferent models and datasets.

Additionally, qualitative evaluation involves visual inspection and interpretation of the segmentation results generated by the imported models. Clinicians and domain experts review the predicted

segmentation maps to assess their clinical relevance, interpretability, and alignment with expert annotations. Qualitative evaluation helps identify potential discrepancies, artifacts, or inaccuracies in the segmentation results and provides valuable feedback for model refinement and improvement.

## 5.8 Ensemble Model:

Develop ensemble fusion strategies to combine predictions from individual models into a final segmentation map.

Experiment with fusion techniques such as majority voting, weighted averaging to optimize segmentation performance.

Each individual model (U-Net, InceptionResNetv2, and WNet) generates segmentation predictions for a given input MRI image. These predictions represent the likelihood or probability of each pixel belonging to different subregions of the glioma, such as enhancing tumor, non-enhancing tumor, necrotic core, and peritumoral edema. The predictions from all individual models are aggregated using the average weighting technique.For each pixel in the segmentation map, the predicted probabilities from corresponding pixels in the segmentation maps generated by U-Net, InceptionResNetv2, and WNet are averaged.

**Weighting Scheme:**

In the average weighting technique, all individual models are given equal weight when computing the ensemble prediction.

Each model's prediction contributes equally to the final segmentation map, regardless of its performance or confidence level.

This simple weighting scheme assumes that all models are equally reliable and can contribute complementary information to improve segmentation accuracy.

**Final Segmentation Map:**

The ensemble prediction generated by averaging the predictions from all individual models represents the final segmentation map for the input MRI image.

Each pixel in the segmentation map indicates the combined confidence or likelihood of belonging to different subregions of the glioma, based on the consensus of all models.

Post-processing Optionally, post-processing steps such as thresholding, morphological operations, or connected component analysis may be applied to the ensemble segmentation map to refine and improve segmentation accuracy.

These post-processing techniques help remove noise, smooth segmentation boundaries, and ensure spatial consistency in the final segmentation results.

## 5.9 Architecture Diagram:

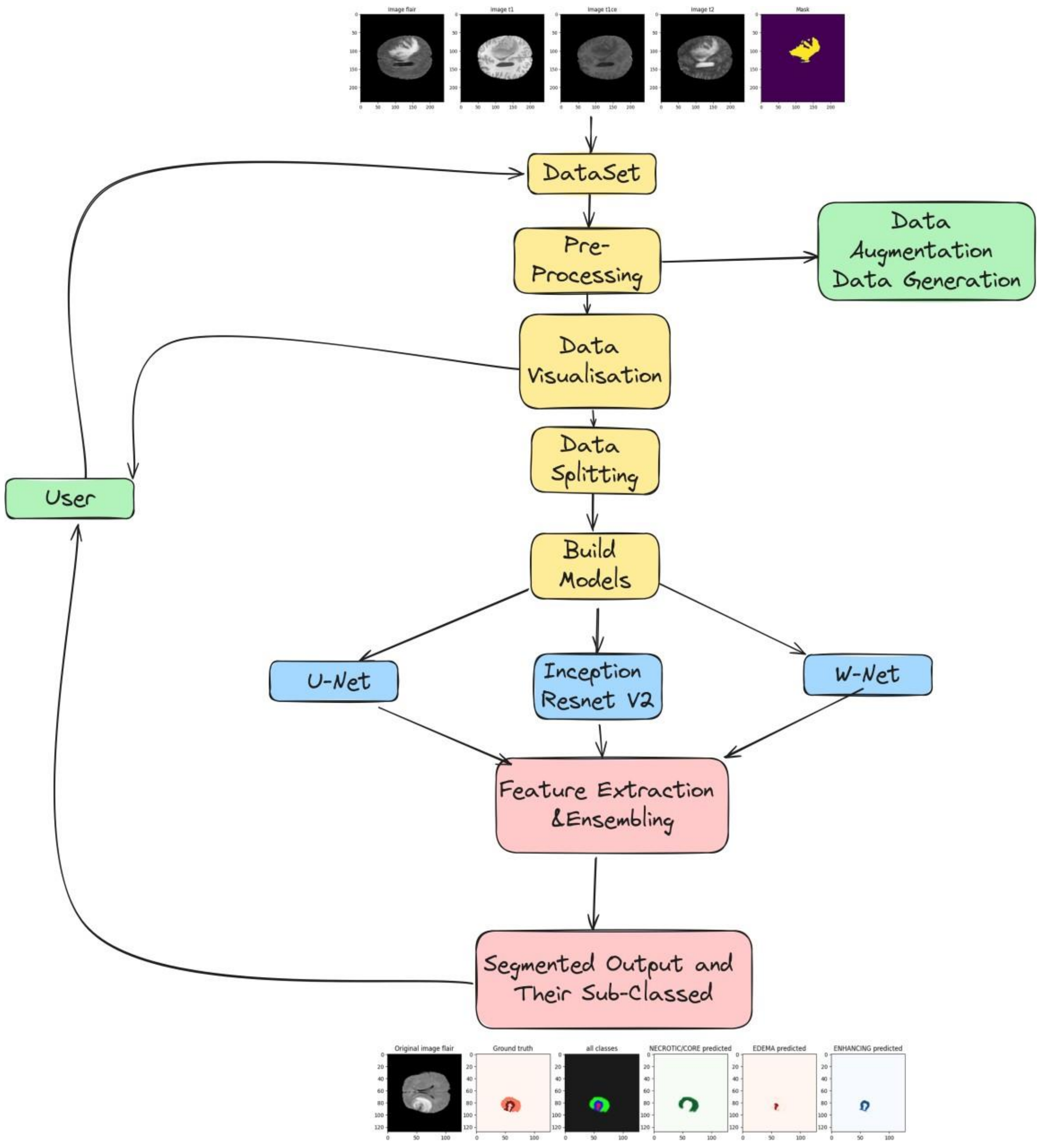


Fig 5.9 Architecture diagram

# 6 . SAMPLE CODE

```
import os
import cv2
import glob
import PIL
import shutil
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from skimage import data
from skimage.util import montage
import skimage.transform as skTrans
from skimage.transform import rotate
from skimage.transform import resize
from PIL import Image, ImageOps

# NEURAL IMAGING
!pip install nilearn
import nilearn as nl
import nibabel as nib
import nilearn.plotting as nlplt
!pip uninstall -y imageio
!pip install imageio==2.27.0
# ML LIBS
from tensorflow import keras
```

```
import keras.backend as K

from keras.callbacks import CSVLogger

import tensorflow as tf

from keras.utils import plot_model

from sklearn.preprocessing import MinMaxScaler

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report

from keras import models

from keras.models import *

from keras import layers

from keras.layers import *

from keras.optimizers import *

from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping,
TensorBoard

from keras.layers.experimental import preprocessing

from keras.metrics import MeanIoU
```

**# Make numpy printouts easier to read.**

```
np.set_printoptions(precision=3, suppress=True)
```

**# DEFINE SEG-AREAS**

```
SEGMENT_CLASSES = {
   0 : 'NOT tumor',
   1 : 'NECROTIC/CORE', # or NON-ENHANCING tumor CORE
   2 : 'EDEMA',
   3 : 'ENHANCING' # original 4 -> converted into 3 later
}
```

```
# there are 155 slices per volume
# to start at 5 and use 145 slices means we will skip the first 5 and last 5
VOLUME_SLICES = 100
VOLUME_START_AT = 22 # first slice of volume that we will include

TRAIN_DATASET_PATH = '/content/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/'
VALIDATION_DATASET_PATH = '/content/brats20-dataset-training-validation/BraTS2020_ValidationData/MICCAI_BraTS2020_ValidationData'

TRAIN_DATASET_PATH = '/content/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/'
VALIDATION_DATASET_PATH = '/content/brats20-dataset-training-validation/BraTS2020_ValidationData/MICCAI_BraTS2020_ValidationData'

test_image_flair=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_flair.nii').get_fdata()
test_image_t1=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_t1.nii').get_fdata()
test_image_t1ce=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_t1ce.nii').get_fdata()
test_image_t2=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_t2.nii').get_fdata()
test_mask=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_seg.nii').get_fdata()

fig, (ax1, ax2, ax3, ax4, ax5) = plt.subplots(1,5, figsize = (20, 10))
slice_w = 25
ax1.imshow(test_image_flair[:,:,test_image_flair.shape[0]//2-slice_w], cmap = 'gray')
ax1.set_title('Image flair')
ax2.imshow(test_image_t1[:,:,test_image_t1.shape[0]//2-slice_w], cmap = 'gray')
ax2.set_title('Image t1')
ax3.imshow(test_image_t1ce[:,:,test_image_t1ce.shape[0]//2-slice_w], cmap = 'gray')
ax3.set_title('Image t1ce')
```

```
ax4.imshow(test_image_t2[:,:,test_image_t2.shape[0]//2-slice_w], cmap = 'gray')
ax4.set_title('Image t2')
ax5.imshow(test_mask[:,:,test_mask.shape[0]//2-slice_w])
ax5.set_title('Mask')
```

**#BUILDING MODEL**

```
def build_unet(inputs, ker_init, dropout):
    conv1 = Conv2D(32, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(inputs)
    conv1 = Conv2D(32, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv1)

    pool = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool)
    conv = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv)

    pool1 = MaxPooling2D(pool_size=(2, 2))(conv)
    conv2 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool1)
    conv2 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv2)

    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool2)
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv3)

    pool4 = MaxPooling2D(pool_size=(2, 2))(conv3)
    conv5 = Conv2D(512, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool4)
    conv5 = Conv2D(512, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv5)
    drop5 = Dropout(dropout)(conv5)

    up7 = Conv2D(256, 2, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling2D(size = (2,2))(drop5))
    merge7 = concatenate([conv3,up7], axis = 3)
    conv7 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge7)
    conv7 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv7)

    up8 = Conv2D(128, 2, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling2D(size = (2,2))(conv7))
    merge8 = concatenate([conv2,up8], axis = 3)
    conv8 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge8)
    conv8 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv8)

    up9 = Conv2D(64, 2, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling2D(size = (2,2))(conv8))
    merge9 = concatenate([conv,up9], axis = 3)
    conv9 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge9)
    conv9 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv9)

    up = Conv2D(32, 2, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling2D(size = (2,2))(conv9))
    merge = concatenate([conv1,up], axis = 3)
    conv = Conv2D(32, 3, activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge)
    conv = Conv2D(32, 3, activation = 'relu', padding = 'same', kernel_initializer =
ker_init)(conv)
    outputs = Conv2D(4, (1,1), activation = 'softmax')(conv)


    model = Model(inputs, outputs, name="U-Net")
    model.compile(loss="categorical_crossentropy",
optimizer=keras.optimizers.Adam(learning_rate=0.001), metrics =
['accuracy',tf.keras.metrics.MeanIoU(num_classes=4), dice_coef, precision, sensitivity,
specificity, dice_coef_necrotic, dice_coef_edema ,dice_coef_enhancing] )
    return model
input_layer = Input((IMG_SIZE, IMG_SIZE, 2))
model_unet = build_unet(input_layer, 'he_normal', 0.2)


IMG_SIZE = 128
```

## DATA GENERATION

```
class DataGenerator(keras.utils.Sequence):
    'Generates data for Keras'
    def___init_(self, list_IDs, dim=(IMG_SIZE,IMG_SIZE), batch_size = 1, n_channels =
2, shuffle=True):
        'Initialization'
        self.dim = dim
        self.batch_size = batch_size
        self.list_IDs = list_IDs
        self.n_channels = n_channels
        self.shuffle = shuffle
        self.on_epoch_end()


    def___len_(self):
        'Denotes the number of batches per epoch'
        return int(np.floor(len(self.list_IDs) / self.batch_size))
```

```
def__getitem_(self, index):
    'Generate one batch of data'
    # Generate indexes of the batch
    indexes = self.indexes[index*self.batch_size:(index+1)*self.batch_size]
    # Find list of IDs
    Batch_ids = [self.list_IDs[k] for k in indexes]
    # Generate data
    X, y = self._data_generation(Batch_ids)
    return X, y
def on_epoch_end(self):
    'Updates indexes after each epoch'
    self.indexes = np.arange(len(self.list_IDs))
    if self.shuffle == True:
        np.random.shuffle(self.indexes)
def__data_generation(self, Batch_ids):
    'Generates data containing batch_size samples' # X : (n_samples, *dim, n_channels)
    # Initialization
    X = np.zeros((self.batch_size*VOLUME_SLICES, *self.dim, self.n_channels))
    y = np.zeros((self.batch_size*VOLUME_SLICES, 240, 240))
    Y = np.zeros((self.batch_size*VOLUME_SLICES, *self.dim, 4))
    # Generate data
    for c, i in enumerate(Batch_ids):
        case_path = os.path.join(TRAIN_DATASET_PATH, i)

        data_path = os.path.join(case_path, f'{i}_flair.nii');
        flair = nib.load(data_path).get_fdata()

        data_path = os.path.join(case_path, f'{i}_t1ce.nii');
        ce = nib.load(data_path).get_fdata()
```

```
            data_path = os.path.join(case_path, f'{i}_seg.nii');
            seg = nib.load(data_path).get_fdata()


            for j in range(VOLUME_SLICES):
                X[j +VOLUME_SLICES*c,:,:,0] =
cv2.resize(flair[:,:,j+VOLUME_START_AT], (IMG_SIZE, IMG_SIZE));
                X[j +VOLUME_SLICES*c,:,:,1] = cv2.resize(ce[:,:,j+VOLUME_START_AT],
(IMG_SIZE, IMG_SIZE));


                y[j +VOLUME_SLICES*c] = seg[:,:,j+VOLUME_START_AT];
        # Generate masks
        y[y==4] = 3;
        mask = tf.one_hot(y, 4);
        Y = tf.image.resize(mask, (IMG_SIZE, IMG_SIZE));
        return X/np.max(X), Y
training_generator = DataGenerator(train_ids)
valid_generator = DataGenerator(val_ids)
test_generator = DataGenerator(test_ids)


def predictByPath(case_path, case, model="InceptionResNetv2Unet"):
    files = next(os.walk(case_path))[2]
    X = np.empty((VOLUME_SLICES, IMG_SIZE, IMG_SIZE, 2))
  #  y = np.empty((VOLUME_SLICES, IMG_SIZE, IMG_SIZE))

    vol_path = os.path.join(case_path, f'BraTS20_Training_{case}_flair.nii');
    flair=nib.load(vol_path).get_fdata()

    vol_path = os.path.join(case_path, f'BraTS20_Training_{case}_t1ce.nii');
    ce=nib.load(vol_path).get_fdata()

 #  vol_path = os.path.join(case_path, f'BraTS20_Training_{case}_seg.nii');
 #  seg=nib.load(vol_path).get_fdata()


    for j in range(VOLUME_SLICES):
        X[j,:,:,0] = cv2.resize(flair[:,:,j+VOLUME_START_AT], (IMG_SIZE,IMG_SIZE))
        X[j,:,:,1] = cv2.resize(ce[:,:,j+VOLUME_START_AT], (IMG_SIZE,IMG_SIZE))
 #       y[j,:,:] = cv2.resize(seg[:,:,j+VOLUME_START_AT], (IMG_SIZE,IMG_SIZE))
  # model.evaluate(x=X,y=y[:,:,:,0], callbacks= callbacks)
    if model == "unet":
      print("Predicting using UNET .... ")
      pred = unet_model.predict(X/np.max(X), verbose=1)
    elif model == "ensemble":
      print("Predicting using Ensemble Model….. ")
      pred = ensemble_model.predict(X/np.max(X), verbose=1)
    else:
      print("Predicting using InceptionResNetv2-UNet. ....")
      pred = InceptionResNetv2_unet_model.predict(X/np.max(X), verbose=1)
    return pred
def showPredictsById(case, start_slice = 60, model="InceptionResNetv2Unet"):
    path = f"/content/brats20-dataset-training-
validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Traini
ng_{case}"
    gt = nib.load(os.path.join(path, f'BraTS20_Training_{case}_seg.nii')).get_fdata()
    origImage = nib.load(os.path.join(path,
f'BraTS20_Training_{case}_flair.nii')).get_fdata()
    p = predictByPath(path, case, model)


    core = p[:,:,:,1]        # GREEN
    edema= p[:,:,:,2]        # RED
    enhancing = p[:,:,:,3]    # BLUE


    plt.figure(figsize=(18, 50))
    f, axarr = plt.subplots(1,6, figsize = (18, 50))
```

```
    for i in range(6): # for each image, add brain background
        axarr[i].imshow(cv2.resize(origImage[:,:,start_slice+VOLUME_START_AT], (IMG_SIZE, IMG_SIZE)), cmap="gray", interpolation='none')


    axarr[0].imshow(cv2.resize(origImage[:,:,start_slice+VOLUME_START_AT], (IMG_SIZE, IMG_SIZE)), cmap="gray")
    axarr[0].title.set_text('Original image flair')
    curr_gt=cv2.resize(gt[:,:,start_slice+VOLUME_START_AT], (IMG_SIZE, IMG_SIZE), interpolation = cv2.INTER_NEAREST)
    axarr[1].imshow(curr_gt, cmap="Reds", interpolation='none', alpha=0.3) # ,alpha=0.3,cmap='Reds'
    axarr[1].title.set_text('Ground truth')
    axarr[2].imshow(p[start_slice,:,:,1:4], cmap="Reds", interpolation='none', alpha=0.3)
    axarr[2].title.set_text('all classes')
    axarr[3].imshow(edema[start_slice,:,:], cmap="Greens", interpolation='none', alpha=0.3)
    axarr[3].title.set_text(f'{SEGMENT_CLASSES[1]} predicted')
    axarr[4].imshow(core[start_slice,:,], cmap="Reds", interpolation='none', alpha=0.3)
    axarr[4].title.set_text(f'{SEGMENT_CLASSES[2]} predicted')
    axarr[5].imshow(enhancing[start_slice,:,], cmap="Blues", interpolation='none', alpha=0.3)
    axarr[5].title.set_text(f'{SEGMENT_CLASSES[3]} predicted')
    plt.show()
    print(f"\n", "-" * 150)
showPredictsById(case=test_ids[11][-3:], model="unet")
showPredictsById(case=test_ids[11][-3:], model="ensemble")
showPredictsById(case=test_ids[11][-3:],model="WNet")
```

# 7.TESTING

## Test Cases:-

Test cases for the segmentation of glioma tumors using a deep learning model, such as U Net or InceptionResNetV2, involve providing various input examples and evaluating the model's performance in segmenting different classes of subregions within the tumor. Here's a short description:

Test Case Description for Glioma Tumor Segmentation:

Input:
Multimodal MR Images: The input includes multiple MRI modalities, such as T1 weighted, T2 weighted, FLAIR, and T1 contrast images. These modalities provide comprehensive information about the tumor's structure and surrounding brain tissue.

Different Classes of Subregion Segmentation:
1. Whole Tumor (WT): Includes all abnormal tissue regions visible in any MRI modality. This is the broadest segmentation class, encompassing both the tumor core and surrounding edema.
2. Tumor Core (TC): Refers to the solid part of the tumor and the enhancing tumor region but excludes the surrounding edema.
3. Enhancing Tumor (ET): Represents the active, enhancing part of the tumor, which appears hyperintense on T1 contrast images.

Explanation:
Test Case 1: Input consists of multimodal MR images of a patient with a known glioma. The expected output is the accurate segmentation of the whole tumor, tumor core, and enhancing tumor regions.

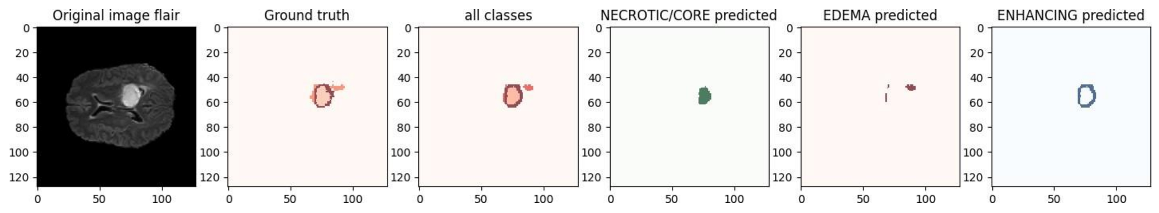


Fig 7.1 TestCase-1

Test Case 2: Similar input from a different patient, but with a smaller and more irregular shaped tumor. This tests the model's ability to handle variations in tumor size and shape.

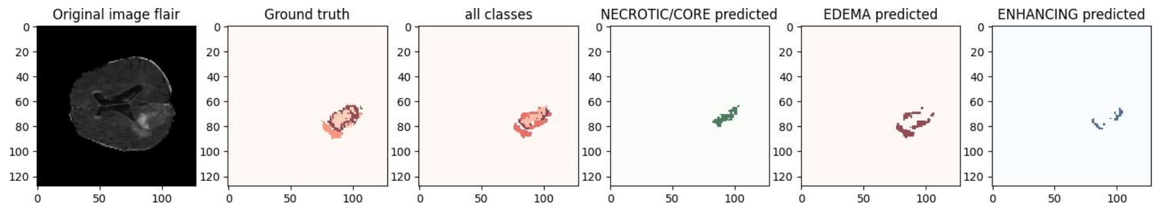


Fig 7.2 TestCase-2

Test Case 3: Input includes MR images with significant noise or artifacts. This evaluates the model's robustness and ability to segment subregions accurately despite image quality issues.

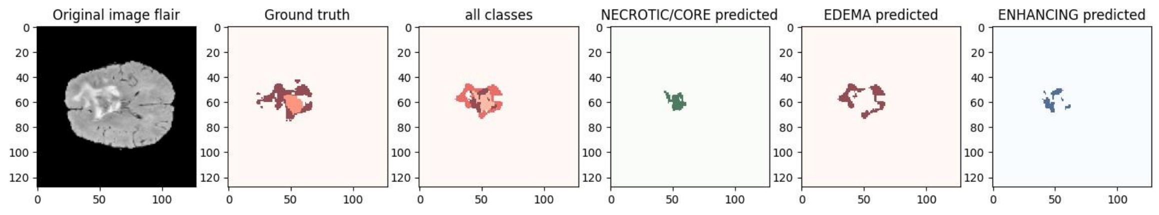


Fig 7.3 TestCase-3

Test Case 4: Input features a glioma with significant edema, challenging the model to distinguish between the tumor core and surrounding affected tissues.

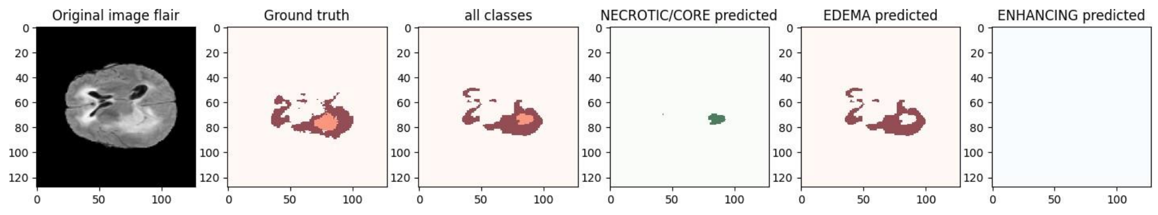


Fig 7.4 TestCase-4

Test Case 5: Input from a follow up scan of a patient post treatment, assessing the model's ability to detect residual or recurring tumor regions.

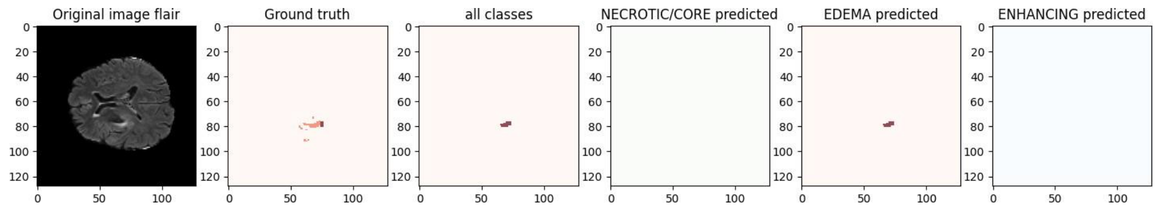


Fig 7.5 TestCase-5

These test cases help ensure that the model accurately identifies and segments the different glioma subregions across various conditions, providing critical information for medical diagnosis and treatment planning.

# 8. RESULTS

**<u>Output Screens:</u>**

## 8.1 Predicting using Unet:

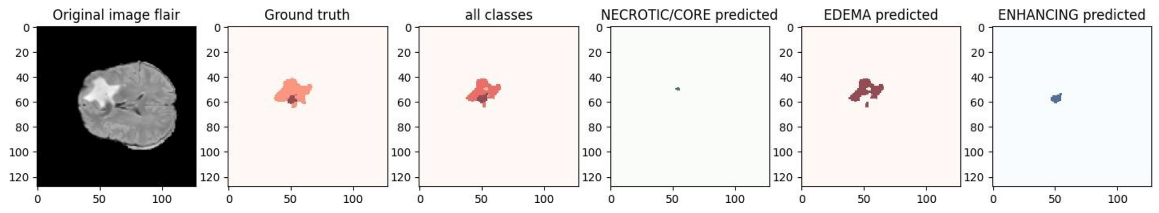


Fig 8.1 Predicting using Unet

## 8.2 Predicting using Inception InceptionResnetV2 V2:

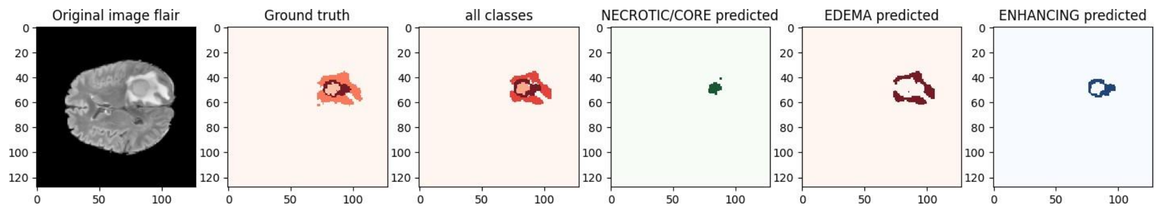


Fig 8.2 Predicting using InceptionResnetV2

## 8.3 Predicting using W-Net:

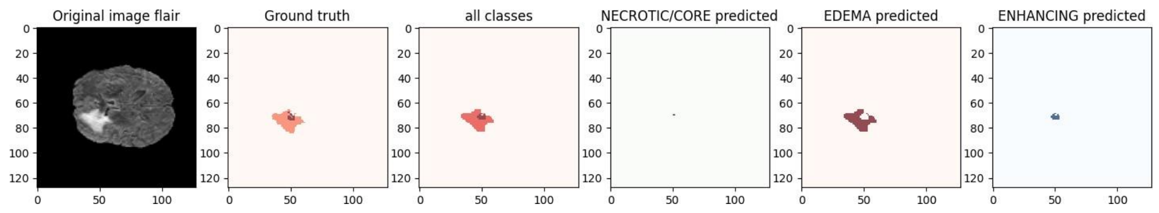


Fig 8.3 Predicting using Wnet

## 8.4 Predicting using Ensemble Model:

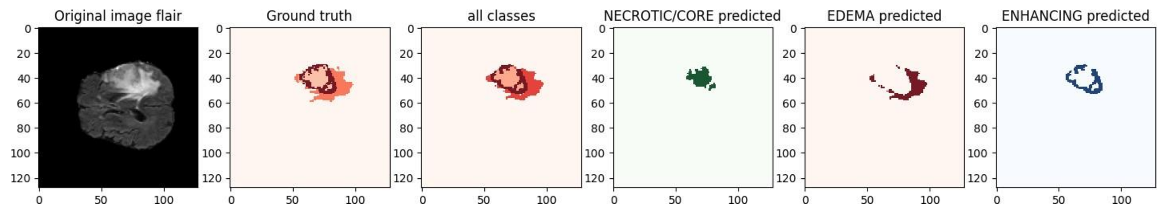
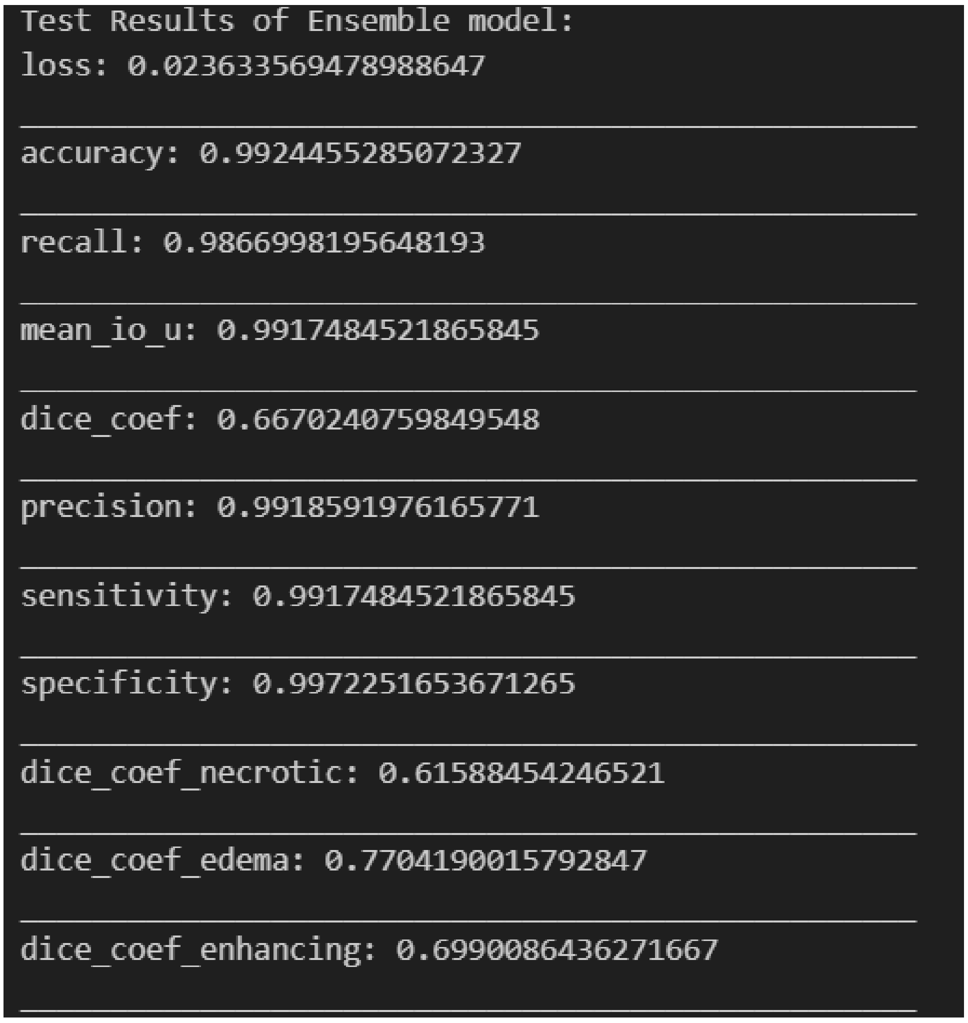


Fig 8.4 Predicting using Ensemble Model



```
Test Results of Ensemble model:
loss: 0.023633569478988647
_______________________________________________
accuracy: 0.9924455285072327
_______________________________________________
recall: 0.9866998195648193
_______________________________________________
mean_io_u: 0.9917484521865845
_______________________________________________
dice_coef: 0.6670240759849548
_______________________________________________
precision: 0.9918591976165771
_______________________________________________
sensitivity: 0.9917484521865845
_______________________________________________
specificity: 0.9972251653671265
_______________________________________________
dice_coef_necrotic: 0.61588454246521
_______________________________________________
dice_coef_edema: 0.7704190015792847
_______________________________________________
dice_coef_enhancing: 0.6990086436271667
_______________________________________________
```

Fig 8.5 Evaluation Metrices of Ensembling Model

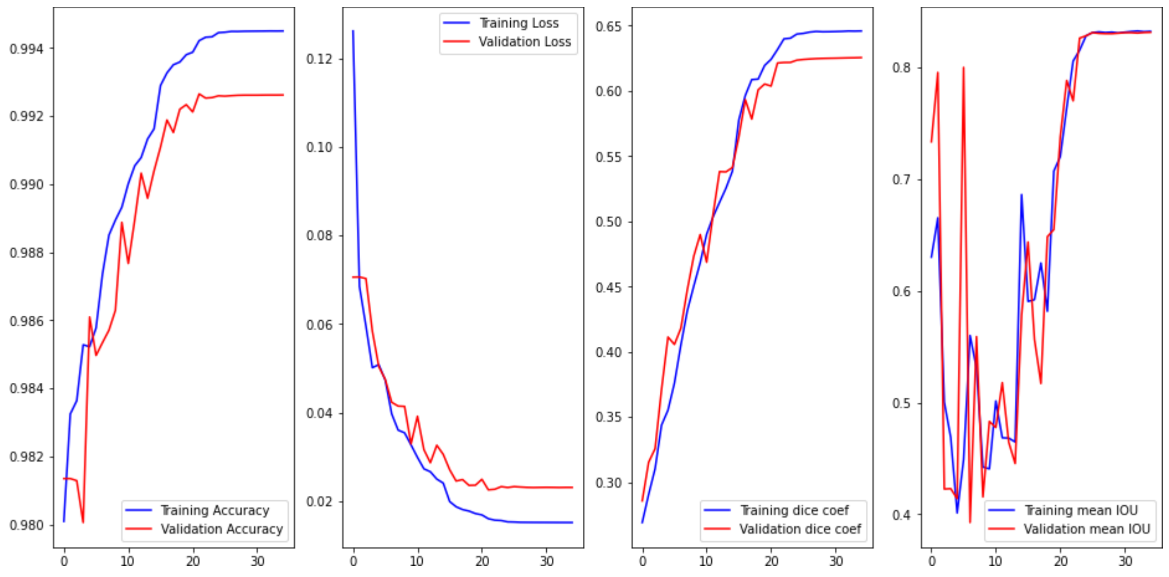


Fig 8.6 Graphical Representation of Metrices (Accuracy , Losses , Dice Coeff , Mean IOU)of the Ensembling Model

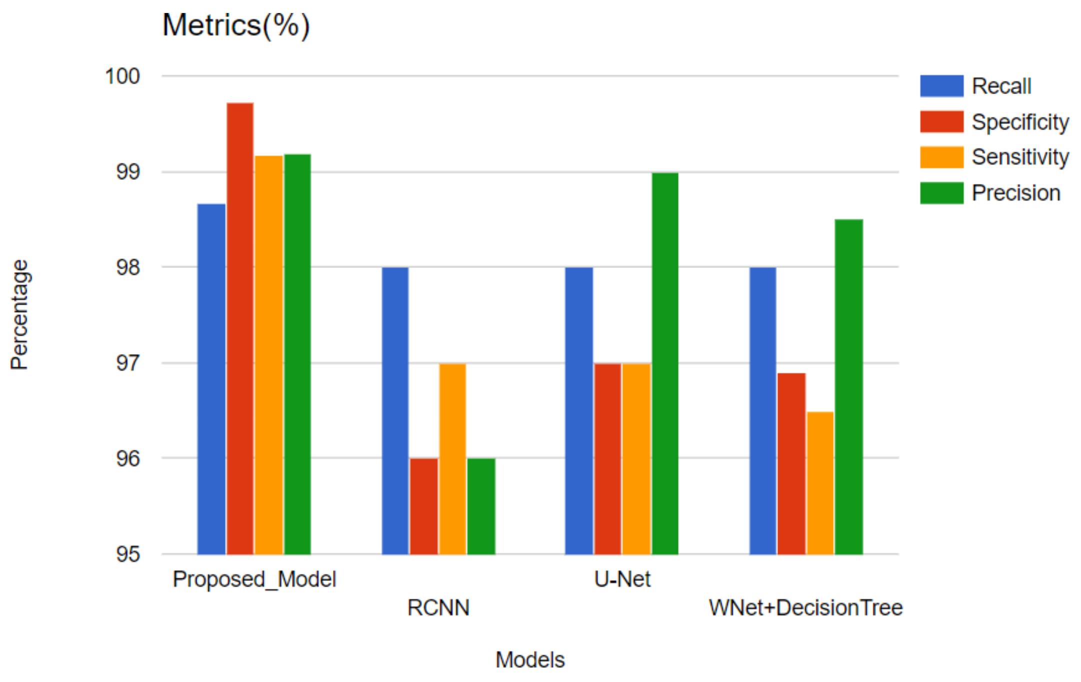


Fig 8.7 Comparision of Metrices(Recall , Specificity , Sensitivity , Precision) with other models

# 9.CONCLUSION

In conclusion, our project introduces a novel approach to glioma subregion segmentation through the integration of deep learning ensembling techniques. By combining the strengths of U-Net, InceptionResNetv2 and WNet architectures. We address the challenge of segmenting glioma subregions into four distinct types with enhanced accuracy and robustness.

The utilization of average prediction and stacking methods for ensembling allows us to harness the diverse predictions of both models, leading

to more comprehensive and precise segmentation outcomes. Through this innovative approach, we aim to advance the field of neuro-oncology by providing clinicians with powerful tools for accurate diagnosis and treatment planning, ultimately improving patient care and outcomes in the management of gliomas.

## FUTURE SCOPE

The project on glioma subregion segmentation using an ensemble of deep learning models offers substantial future potential. Enhanced clinical translation involves validating models in diverse patient cohorts, leading to integration into clinical workflows for improved treatment planning. Multicenter collaborations can validate model generalizability across varied imaging protocols and patient demographics. Future research may prioritize model interpretability, employing explainable AI techniques to elucidate segmentation decisions. Longitudinal analysis can track glioma progression and treatment response over time, guiding personalized interventions. Exploring advanced imaging modalities could enrich model inputs, refining subregion delineation accuracy. Integration into clinical decision support systems empowers clinicians with quantitative insights for optimal patient care. Overall, the project's future scope encompasses clinical validation, interdisciplinary collaboration, and advancements in model interpretability, multi-modal integration, and clinical translation, promising significant contributions to neurooncology practice and research .

# 10. REFERENCES

[1] K. Farmanfarma, M. Mohammadian, Z. Shahabinia, S. Hassanipour, and H.Salehiniya, "Brain cancer in the world: an epidemiological review," WorldCancer Res. J., vol. 6, no. 5, pp. 1–5, 2019. https://doi.org/10.32113/wcrj_20197_1356

[2] D. N. George, H. B. Jehlol, and A. S. Oleiwi, "Brain tumor detection using shapefeatures and machine learning algorithms," Int. J. Adv. Res. Computer Sci. Softw.Eng., vol. 5, no. 10, pp. 454–459, 2015.

[3] D. N. Louis, A. Perry, G. Reifenberger, A. Von Deimling, D. Figarella-Branger,W. K. Cavenee, et al., "The 2016 World Health Organization classification of tumors of the central nervous system: a summary," Acta Neuropathol., vol. 131,no. 6, pp. 803–820, 2016. https://doi.org/ 10.1007/s00401-016-1545-1

[4] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic brain tumor detectionand segmentation using U-Net based fully convolutional networks," in: Annual Conference on Medical Image Understanding and Analysis, Springer, 2017, pp. 506–517. https://doi.org/ 10.1007/978-3-319-60964-5_44

[5] S.Das, S. Bose, G. K. Nayak, S. C. Satapathy, and S. Saxena, "Brain tumor segmentation and overall survival period prediction in glioblastoma multiforme using radiomic features," Concurrency Comput.: Pract. Experience, p. e6501. https://doi.org/ 10.1002/cpe.6501

[6] S.Das, "Brain Tumor Segmentation from MRI Images Using Deep Learning Framework," in: Progress in Computing, Analytics and Networking, Springer, Singapore, 2020, pp. 105–114. https://doi.org/ 10.1007/978-981-15-2414-1_11

[7] S. Das, G. Nayak, S. Saxena, S. C. Satpathy, "Effect of learning parameters onthe performance of UNet Model in segmentation of Brain tumor," Multimed. Tools Appl., pp. 1–19, 2021. https://doi.org/ 10.1007/s11042-021-11273-5

[8] S. Saxena, P. Mohapatra, and S. Pattnaik, "Brain tumor and its segmentation from brain MRI sequences," in: Early Detection of Neurological Disorders UsingMachine Learning Systems, IGI Global, 2019, pp. 39–60. https://doi.org/ 10.4018/978-1-5225-8567-1.ch004

[9] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, et al., "The multimodal brain tumor image segmentation benchmark (BRATS)," IEEE Trans. Med.

Imaging, vol. 34, no. 10, pp. 1993–2024, Oct 2015. https://doi.org/ 10.1109/TMI.2014.2377694

[10] A.Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv. Neural Inf. Process. Syst., vol. 25, pp.1097–1105, 2012. https://doi.org/ 10.1145/3065386

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition, 2014, pp. 580– 587. https://doi.org/10.1109/CVPR.2014.81

[12] D. Yi, M. Zhou, Z. Chen, and O. Gevaert, “3-D convolutional neural networks for glioblastoma segmentation,” arXiv preprint arXiv:1611.04534,2016.

[13] M. El Adoui, S. A. Mahmoudi, M. A. Larhmam, and M. Benjelloun, “MRI breast tumor segmentation using different encoder and decoder CNN architectures,” Computers, vol. 8, no. 3, p. 52, 2019. https://doi.org/ 10.3390/computers8030052

[14] S. Alqazzaz, X. Sun, X. Yang, and L. Nokes, “Automated brain tumor segmentation on multi-modal MR image using WNet,” Comput. Vis. Media, vol.5, no. 2, pp. 209–219, 2019. https://doi.org/ 10.1007/s41095-019-0139-y

[15] G.R. Padalkar and M. B. Khambete, “Analysis of Basic-WNet architecture with variations in training options,” in: International Conference on IntelligentSystems Design and Applications, Springer, Vellore, India, 2018, pp. 727–735. a. https://doi.org/ 10.1007/978-3-030-16657-1_68

[16] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks forbiomedical image segmentation,” in: International Conference on Medical image b. Computing And Computer-Assisted Intervention, Springer, Munich, Germany, 2015, pp. 234–241. https://doi.org/ 10.1007/978-3-319-24574- 4_28

[17] F. Xu, H. Ma, J. Sun, R. Wu, X. Liu, and Y. Kong, “LSTM multi-modal UNetfor brain tumor segmentation,” in: 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), IEEE, Xiamen, China, 2019, pp. 236–240. https://doi.org/ 10.1109/ICIVC47709.2019.8981027

[18] S. Saxena, S. Paul, A. Garg, A. Saikia, and A. Datta, “Deep Learning in Computational

Neuroscience," in: Challenges and Applications for Implementing Machine Learning in Computer Vision, IGI Global, 2020, pp. 43–63. https://doi.org/ 10.4018/978-1-7998-0182-5.ch002

[19] S. Das, M. K. Swain, G. Nayak, and S. Saxena, "Brain tumor segmentation from 3D MRI slices using cascading convolutional neural network," in: Advances in Electronics, Communication and Computing, Springer, Bhubaneswar, India, 2021, pp. 119–126. https://doi.org/10.1007/978-981-15-8752-8_12

[20] Rajbdad, M. Aslam, S. Azmat, T. Ali, and S. Khattak, "Automated fiducial points detection using human body segmentation," Arab. J. Sci. Eng., vol. 43, no.2, pp. 509–524, 2018. https://doi.org/ 10.1007/s13369-017-2646-4

[21] A.Demirhan, M. Törü, and I. Güler, "Segmentation of tumor and edema along with healthy tissues of brain using wavelets and neural networks," IEEE journal of biomedical and health informatics, vol. 9, no. 4, pp. 1451–1458, 2014.10.1109/JBHI.2014.2360515 https://doi.org/10.1109/JBHI.2014.2360515

[22] M. Lyksborg, O. Puonti, M. Agn, and R. Larsen, "An ensemble of 2Dconvolutional neural networks for tumor segmentation," in: Scandinavianconference on image analysis, Springer, Cham, pp. 201–211, 2015. https://doi.org/10.1007/978-3-319-19665-7_17

[23] S. Furqan Qadri, D. Ai, G. Hu, M. Ahmad, Y. Huang, Y. Wang, et al.,"Automatic deep feature learning via patch-based deep belief network forvertebrae segmentation in CT images," Appl. Sci., vol. 9, no. 1, p. 69, 2019. https://doi.org/10.3390/app9010069.

[24] K. Kamnitsas, W. Bai, E. Ferrante, S. McDonagh, M. Sinclair, N. Pawlowski,et al., "Ensembles of multiple models and architectures for robust brain tumour segmentation," in: International MICCAI Brainlesion Workshop, IEEE, Canada,2017, pp. 450–462. https://doi.org/10.1007/978-3-319-75238-9_38.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in: Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.https:/doi.org/ 10.1109/CVPR.2015.7298965.