

Segmentation of Gliomas in Pre-Operative MRI Scans using U-Net Architecture

Contents

1	Introduction to Brain Cancer	1
1.1	Types of Brain Tumours	2
1.2	Gliomas	3
1.2.1	Treatment of Gliomas	3
1.2.2	Need for Detection	5
2	Introduction to the Problem Statement	6
2.1	Tumor Sub-Regions	6
2.2	Labeling MRI Pixels	7
2.3	Objective and Importance	8
2.4	Approach to Solving the Problem Statement	10
3	Dataset Description	11
3.1	MRI Modalities in BraTS Dataset	13
3.2	Data Acquisition and Pre-Processing	14
3.2.1	Data Sources	14
3.2.2	Pre-Processing	14
3.3	Segmentation Annotations	14
3.4	Pre-Processing for Training Model	15
3.4.1	Defining Segmentation Areas (<code>SEGMENT_CLASSES</code>)	15
3.4.2	Volume Slices Selection Parameters	15
3.4.3	Dataset Paths	16
3.4.4	Loading MRI Modalities	16
4	U-Net Architecture	18
4.1	Architecture Overview	18
4.1.1	Contracting Path (Encoder)	19

4.1.2	Bottleneck	19
4.1.3	Expanding Path (Decoder)	19
4.1.4	Skip Connections	19
4.1.5	Final Layer	20
4.2	Loss Function	20
4.3	Advantages	20
4.4	Applications	20
4.5	Extensions and Variants	21
5	Performance Metrics	23
5.1	Loss Function : Dice Coefficient	23
5.2	Precision	23
5.3	Sensitivity	24
5.4	Recall	24
6	Loading the Dataset	25
6.1	Data Generators	25
6.1.1	Benefits of Using Data Generators	25
6.2	Data Split	26
6.2.1	Splitting the Dataset	26
7	Model Performance	28
8	Predictions	29
8.1	Code and Resources	30

1. Introduction to Brain Cancer

Brain cancer arises when cells within the brain begin to grow uncontrollably, forming a mass of abnormal cells known as a tumor. Brain tumors are classified into two main categories: primary brain tumors and secondary (metastatic) brain tumors. Primary brain tumors originate in the brain itself, whereas secondary brain tumors begin elsewhere in the body and spread to the brain. Brain cancer can be benign (non-cancerous) or malignant (cancerous), with malignant brain tumors being more aggressive and likely to grow quickly, invading surrounding tissues.

The causes of brain cancer are not fully understood. However, genetic mutations and environmental factors like exposure to radiation or certain chemicals are known to increase the risk. Brain cancer affects cognitive and motor functions due to its impact on critical brain areas, making it one of the more serious and challenging cancers to treat.

This report shall focus mainly with primary tumours, specifically, gliomas.

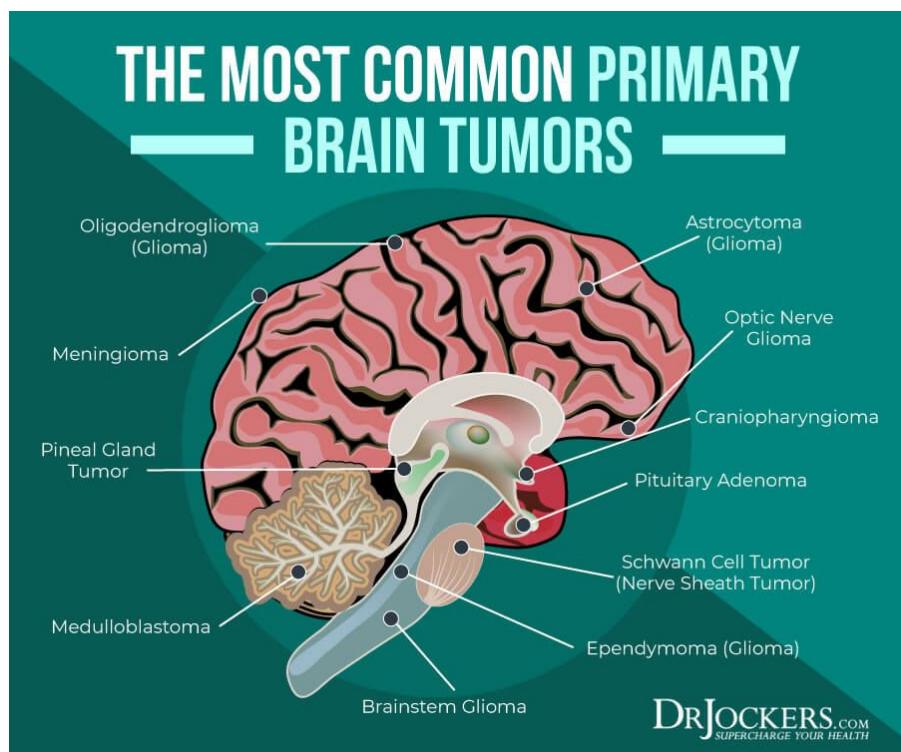


Figure 1.1: Overview of Primary Brain Tumours

1.1 Types of Brain Tumours

Brain tumors are classified based on the type of cells from which they arise, with several subtypes including:

- **Gliomas:** These are tumors that arise from glial cells, the supportive cells of the brain. Gliomas are the most common type of primary brain tumor, accounting for approximately 30% of all brain tumors and about 80% of malignant brain tumors.
- **Meningiomas:** Tumors that originate in the meninges, the protective layers surrounding the brain and spinal cord. Most meningiomas are benign.
- **Pituitary Tumors:** These tumors occur in the pituitary gland at the base of the brain and can disrupt hormone levels, impacting various body functions.
- **Medulloblastomas:** Common in children, these tumors originate in the cerebellum and are highly malignant.
- **Ependymomas:** Tumors that develop from ependymal cells lining the ventricles of the brain and the spinal canal. They are more common in children and young adults.

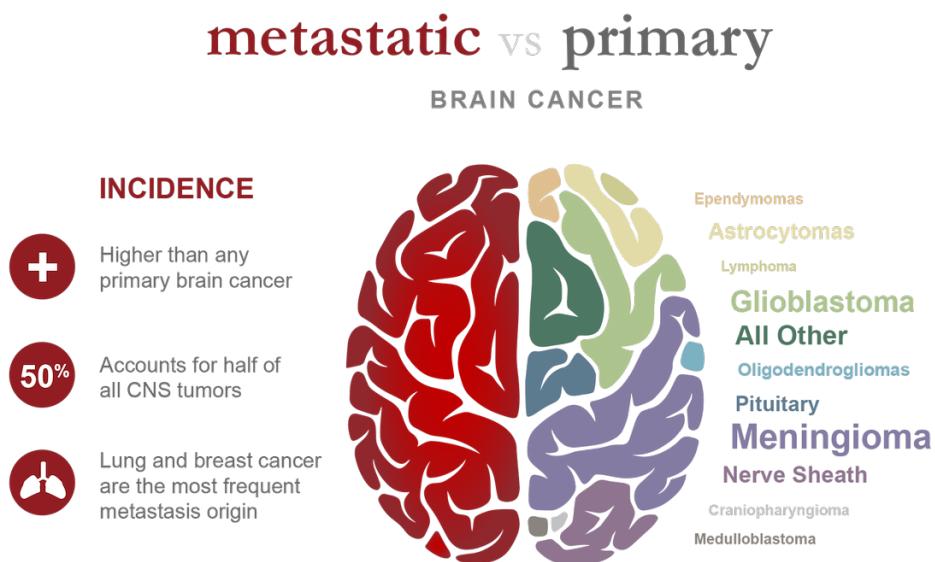


Figure 1.2: Overview of the different types of tumours. Some of the extra types shown in the image are not relevant to our report and thus have been ignored.

In this report, our goal is to focus on using Machine Learning and Deep Learning to try and segment areas of the brain that are affected by a glioma/glioblastoma. Thus, we shall delve into further detail about gliomas in the further sections.

1.2 Gliomas

Gliomas are the most prevalent type of malignant brain tumor and encompass several subtypes based on the specific type of glial cell they affect:

- Astrocytomas: Tumors originating from astrocytes, which are star-shaped glial cells that support and nourish neurons.
- Oligodendrogiomas: Tumors that develop from oligodendrocytes, the cells responsible for producing myelin, the insulating layer around nerves.
- Ependymomas: Although classified under gliomas, these arise from ependymal cells rather than traditional glial cells.

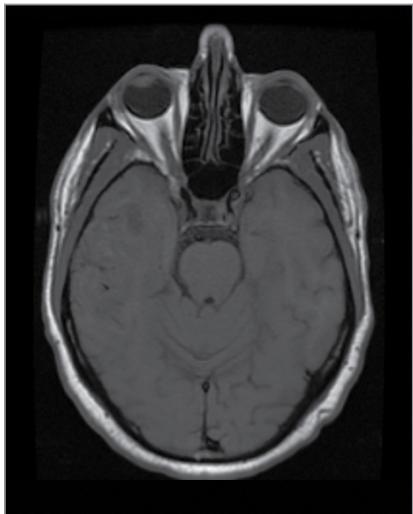
Among gliomas, glioblastoma multiforme (GBM) is the most aggressive and is associated with a poor prognosis. GBM grows quickly and tends to infiltrate surrounding brain tissue, making complete surgical removal challenging. Gliomas are typically graded on a scale of I to IV, where low-grade (I-II) gliomas are slow-growing, and high-grade (III-IV) gliomas are more aggressive, with grade IV being classified as glioblastoma.

1.2.1 Treatment of Gliomas

The treatment of gliomas depends on several factors, including the tumor's type, location, size, and grade, as well as the patient's age and overall health. Standard treatment approaches include:

1. Surgery: Primary treatment involving tumor resection, often aided by intraoperative MRI and neuronavigation to maximize tumor removal and preserve healthy tissue.
2. Radiation Therapy: Used post-surgery, especially for high-grade gliomas, to target resid-

A Axial T1-weighted MRI without contrast



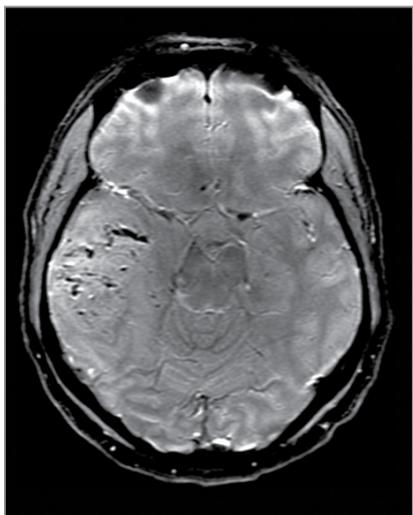
B Axial T1-weighted MRI with gadolinium contrast



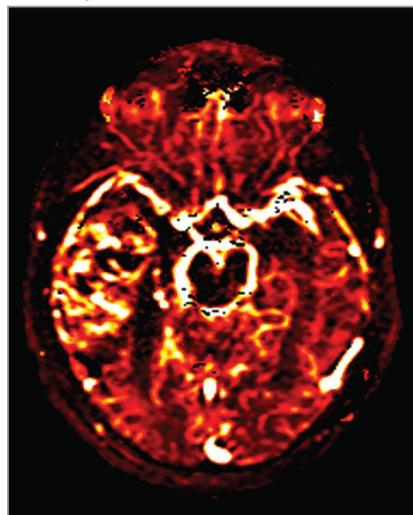
C Fluid attenuated inversion recovery (FLAIR) MRI



D Susceptibility weighted imaging MRI



E DCE T1-weighted plasma volume map



F DCE volume transfer coefficient (Ktrans) map

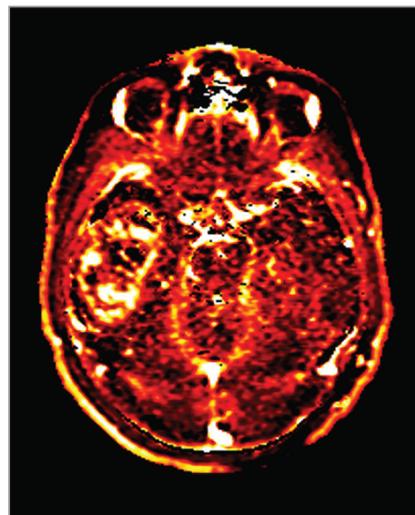


Figure 1.3: Figure above shows the results of various types of MRI scans taken on a patient with a glioma.

ual cells, with precise techniques like stereotactic radiosurgery minimizing damage to nearby tissue.

3. Chemotherapy: Temozolomide is commonly used alongside radiation, crossing the blood-brain barrier but limited by side effects and drug penetration challenges.
4. Targeted Therapy and Immunotherapy: Targeted drugs like bevacizumab and immunotherapies seek to inhibit tumor growth and stimulate immune response, primarily in experimental or recurrent cases.

5. Experimental Therapies: Clinical trials explore gene therapy, tumor-treating fields (TTF), and vaccines, with TTF showing promise in glioblastoma treatment.

1.2.2 Need for Detection

Accurate segmentation of tumorous versus non-tumorous pixels is crucial in brain cancer treatment, as it directly impacts surgical precision. Deep learning (DL) and machine learning (ML) models offer a high degree of accuracy in identifying the exact boundaries of tumors, allowing surgeons to distinguish between malignant and healthy brain tissue. This precise delineation is essential during resection, enabling neurosurgeons to remove as much of the tumor as possible without damaging surrounding healthy tissue. By preserving critical brain areas, such as those responsible for motor and cognitive functions, patients have a better chance of maintaining a higher quality of life post-surgery.

In addition, accurate segmentation is fundamental for effective radiation therapy planning. Radiation treatment aims to target residual cancer cells while minimizing exposure to healthy tissue. With DL/ML segmentation, radiation oncologists can create precise treatment plans that deliver focused doses to tumor areas. This not only improves the chances of killing residual cancer cells but also reduces the likelihood of adverse side effects, which is especially critical for sensitive brain tissues. As a result, patients can receive higher doses targeted specifically at the tumor with reduced risk of collateral damage to vital brain regions.

2. Introduction to the Problem Statement

Gliomas, a type of brain tumor, are complex and heterogeneous, and accurate segmentation of these tumors in MRI scans is crucial for effective diagnosis, treatment planning, and monitoring. The goal of this segmentation task is to classify every pixel in an MRI scan to determine if it belongs to a tumor region, and if so, which part of the tumor it is associated with.

2.1 Tumor Sub-Regions

In glioma segmentation, the tumor is divided into several important sub-regions that are significant for clinical evaluation. These sub-regions provide insights into tumor growth, aggressiveness, and the overall tumor burden. The key sub-regions include:

- **Enhancing Tumor (ET)** : This is the part of the tumor that shows active enhancement with a contrast agent. It is often associated with the most aggressive areas of the tumor, which are typically highly vascularized and growing rapidly. The enhancing tumor (ET) is critical for assessing tumor activity and aggressiveness.
- **Tumor Core (TC)** : The tumor core encompasses the central part of the tumor, including both viable tumor cells and necrotic (dead) tissue, but excluding the peripheral enhancing areas. This region represents the overall size and extent of the tumor, minus the highly active, contrast-enhanced areas.
- **Whole Tumor (WT)** : This includes all areas of the tumor, combining the enhancing tumor, tumor core, and the surrounding edema (swelling). The whole tumor region provides a comprehensive view of the tumor's full extent and burden, which is crucial for overall monitoring and treatment planning.

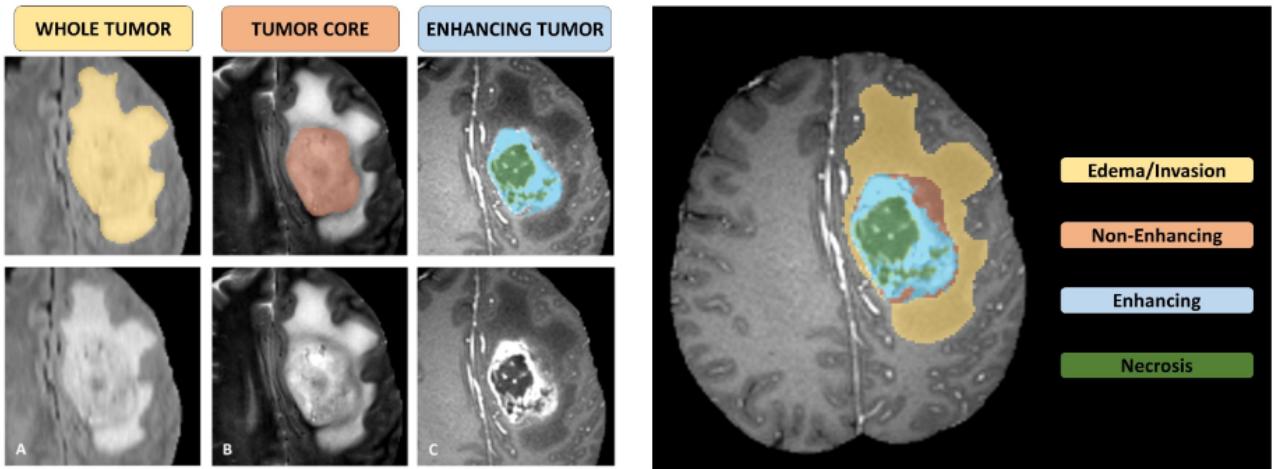


Figure 2.1: The different sub regions in a glioma.

2.2 Labeling MRI Pixels

The next step is to label each pixel in the MRI image according to whether it belongs to any tumor sub-region or represents healthy brain tissue. The labeling scheme for the segmentation is as follows:

- **Label 1: NCR & NET (Non-Enhancing Tumor and Necrotic Tissue)** - This label marks regions of the tumor that do not show enhancement with contrast. These areas might represent less active or necrotic (dead) tissue. It helps identify parts of the tumor that are not actively growing.
- **Label 2: Edema (ED)** - Edema refers to the swelling around the tumor. While not cancerous itself, edema is an important feature that surrounds tumors, often indicating how aggressive a tumor is. It is marked with a value of 2.
- **Label 4: Enhancing Tumor (ET)** - The enhancing tumor (ET) is the region of the tumor that shows up brightly after contrast is applied during an MRI scan. It indicates high metabolic activity and is often associated with more aggressive and fast-growing parts of the tumor. This region is labeled as 4.
- **Label 0: Non-Tumor Areas** - Label 0 is used for all areas that do not belong to the tumor, such as healthy brain tissue or non-tumor regions. These areas are considered to

be free of any abnormal growth or swelling.

Note : Label 3 has been preserved and kept as unused for future use in glioma imaging. But in this experiment, since we have no reason to reserve this label, we shall use Label 3 for Enhancing Tumours in the place of Label 4.

2.3 Objective and Importance

The primary objective of this segmentation task is to assign the appropriate label to each pixel in the MRI image, effectively dividing the scan into tumor and non-tumor regions, as well as categorizing the tumor into its relevant sub-regions (ET, TC, WT). This segmentation has several important clinical applications:

1. **Assessment of Tumor Progression** - By tracking the changes in these tumor sub-regions over time, clinicians can monitor how the tumor is evolving. This includes observing whether it is growing, shrinking, or responding to treatment.
2. **Treatment Planning** - Accurate segmentation helps in planning surgeries or radiotherapy. Knowing the exact location and extent of the tumor, including all active and non-active regions, allows for more precise treatment strategies tailored to the patient's specific tumor characteristics.
3. **Monitoring Patient Response** - Once treatment begins, continuous monitoring of tumor regions (especially the enhancing tumor) is vital to detect signs of recurrence or further progression. Segmentation offers a clear baseline to measure these changes over time.

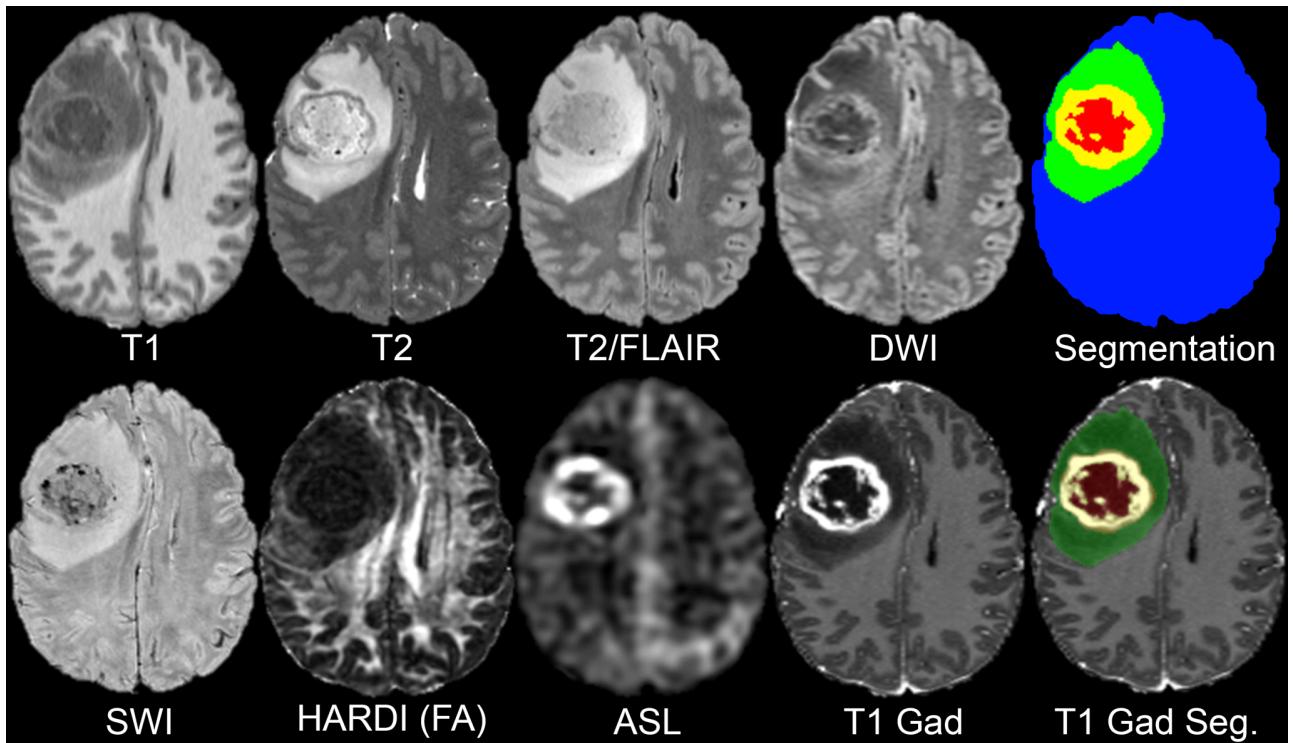


Figure 2.2: The expected output from the segmentation task. The image shows various brain scans using different MRI imaging techniques and finally, the coloured image shows the segmented tumour. The Blue section stands for Healthy Tissue, Green section for Edema, Yellow section for Enhancing Tumour and Red section for Necrotic Tissue.

In summary, precise segmentation of gliomas into distinct sub-regions (Enhancing Tumor, Tumor Core, and Whole Tumor) is essential for understanding the tumor's behavior and planning treatment. This task involves classifying each pixel in an MRI scan into one of several categories: part of the tumor, part of the surrounding edema, or healthy tissue. Automated segmentation methods, often using advanced machine learning algorithms, play a key role in improving accuracy, consistency, and efficiency, ultimately benefiting clinical practice. By utilizing such methods, clinicians can gain a better understanding of the tumor, monitor its progress, and make more informed decisions about treatment.

2.4 Approach to Solving the Problem Statement

To solve the problem of tumour segmentation to detect gliomas, we shall use the following flow of data:

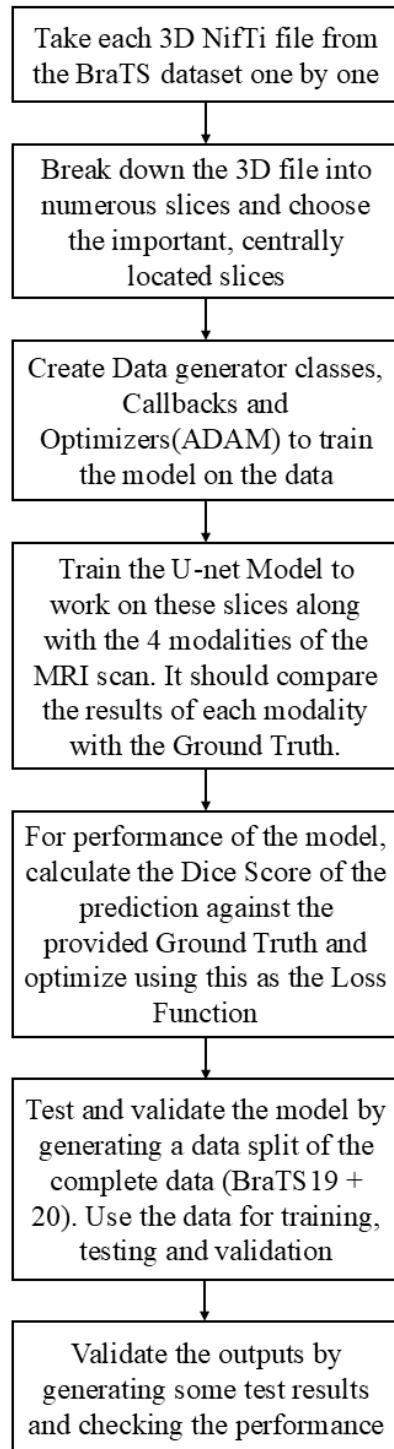


Figure 2.3: Flowchart representing the strategy used to perform the tumour segmentation task.

3. Dataset Description

For the segmentation task, we require a dataset that contains well cleaned and pre-processed MRI data of cancer patients who have a glioma in their brain. For this, the BraTS dataset is a perfect candidate.

The Brain Tumor Segmentation (BraTS) dataset is a widely recognized and comprehensive dataset used extensively for research and development in brain tumor segmentation using machine learning and deep learning techniques. Created as part of the Brain Tumor Segmentation Challenge, the BraTS dataset has become a standard benchmark for evaluating algorithms aimed at accurately segmenting brain tumors, especially gliomas, in medical imaging.

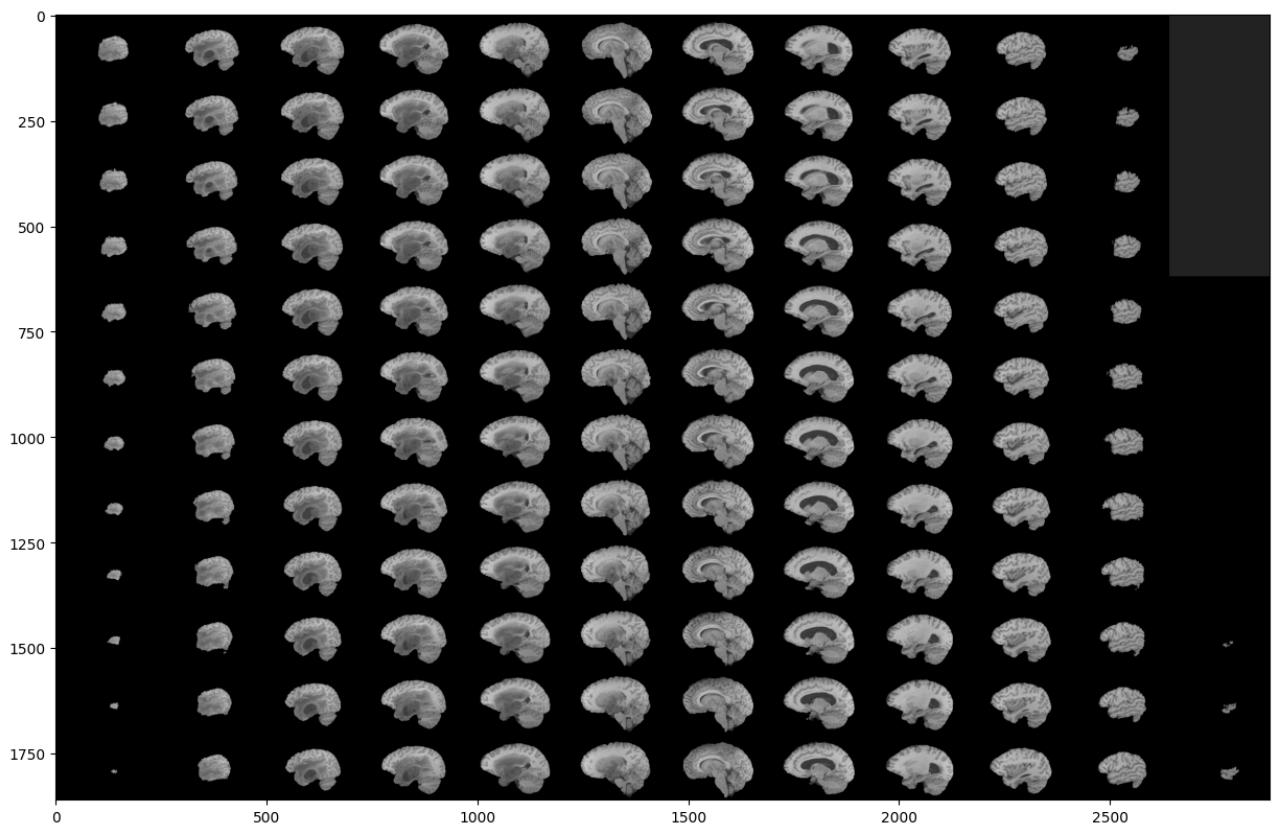


Figure 3.1: Images of cross-sectional views of the 3-D MRI Scan of the brain from BraTS dataset.

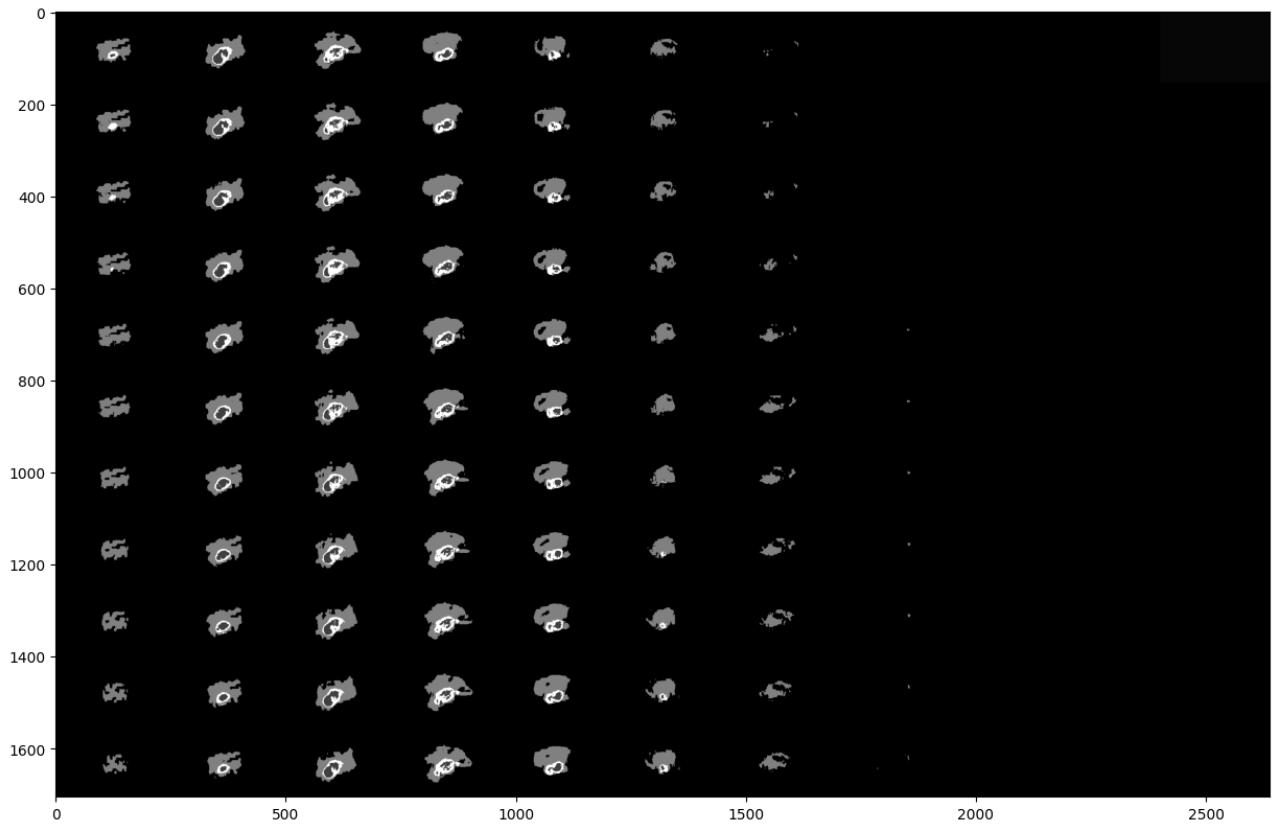


Figure 3.2: Images of cross-sectional views of the 3-D MRI Scan of the glioma of the same patient from BraTS dataset. (Ground Truth)

The BraTS dataset provides high-quality, annotated MRI scans of brain tumor patients, focusing on gliomas, which are the most common and aggressive brain tumors. It includes both low-grade gliomas (LGG) and high-grade gliomas (HGG), allowing for a range of applications in segmentation and classification. The dataset's annotations have been rigorously performed by expert radiologists, ensuring reliable ground truth labels for training and testing.

The BraTS (Brain Tumor Segmentation) dataset provides multimodal MRI brain scans in NIfTI (.nii.gz) format, a common format for medical imaging data storage. These scans capture detailed 3D brain images under various MRI sequences and are pre-processed to a uniform resolution, suitable for brain tumor segmentation research. Below is a description of each modality included in the dataset and the associated segmentation annotations.

3.1 MRI Modalities in BraTS Dataset

Each patient case in the BraTS dataset consists of MRI scans in four different modalities, each highlighting different characteristics of the tumor and surrounding brain tissue. These modalities include:

1. **T1-weighted (T1):**

This modality provides high-resolution images of anatomical structures in the brain, which helps in visualizing brain anatomy but does not highlight tumors effectively.

2. **T1-weighted post-contrast (T1Gd):**

T1 images are enhanced with a gadolinium-based contrast agent, which helps highlight areas of high blood-brain barrier permeability, making it easier to identify the active tumor region.

3. **T2-weighted (T2):**

T2 images highlight water content in the tissue, making it easier to visualize edema, which often surrounds brain tumors. This modality is particularly helpful in identifying both tumorous and non-tumorous fluid regions.

4. **Fluid-attenuated inversion recovery (FLAIR):**

FLAIR images suppress the cerebrospinal fluid signal, enhancing visibility of peritumoral edema, often associated with high-grade gliomas. FLAIR is critical for detecting the tumor boundary, as it captures regions with abnormal fluid signals surrounding the tumor.

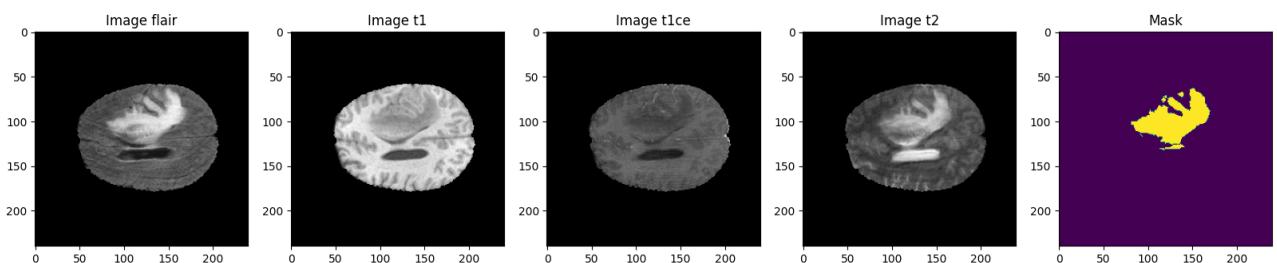


Figure 3.3: Images of the different modalities available from the dataset.

3.2 Data Acquisition and Pre-Processing

3.2.1 Data Sources

The BraTS dataset was collected from 19 different institutions, capturing a diverse range of clinical protocols and MRI scanner types. This diversity in acquisition sources helps ensure that models trained on BraTS data can generalize well across different imaging settings and patient populations.

3.2.2 Pre-Processing

1. **Co-Registration:** Each MRI scan is aligned to a common anatomical template. This process, called co-registration, helps standardize the orientation and positioning of images across patients, making it easier for machine learning models to focus on tumor characteristics rather than variations in patient anatomy.
2. **Interpolation:** All images are resampled to a standard voxel resolution of 1 mm^3 , ensuring uniformity in spatial dimensions. This consistent resolution allows models to analyze images with the same level of detail, improving segmentation accuracy across different scans.
3. **Skull-Stripping:** Extracranial areas, such as the skull and other non-brain tissues, are removed from the images. This process, known as skull-stripping, isolates brain structures, enabling models to focus on relevant tumor regions without interference from surrounding tissues.

3.3 Segmentation Annotations

All imaging datasets are manually segmented by trained raters (one to four per scan) following a standardized annotation protocol. Final annotations are validated by experienced neuro-radiologists. Segmentation labels correspond to distinct tumor regions and include the following:

- GD-Enhancing Tumor (ET): Labeled as 4, this region corresponds to areas of contrast enhancement within the tumor, indicating active tumor tissue.
- Peritumoral Edema (ED): Labeled as 2, this region shows swelling around the tumor, often due to fluid accumulation.
- Necrotic and Non-Enhancing Tumor Core (NCR/NET): Labeled as 1, representing the dead (necrotic) tissue.

3.4 Pre-Processing for Training Model

3.4.1 Defining Segmentation Areas (`SEGMENT_CLASSES`)

The `SEGMENT_CLASSES` dictionary is used to map integer labels to specific types of brain tissue or areas for a medical imaging segmentation task. Each key represents a segmentation label, and each value is a description of the corresponding tissue class:

- **0: NOT tumor** – Represents non-tumor tissue.
- **1: NECROTIC/CORE** – Represents the necrotic or non-enhancing tumor core.
- **2: EDEMA** – Represents regions with edema (swelling around the tumor).
- **3: ENHANCING** – Represents enhancing tumor areas (often converted from a different label, originally labeled as 4).

3.4.2 Volume Slices Selection Parameters

- **VOLUME_SLICES**: Set to 100, indicating that each 3D file shall be split into 100 slices. The imaging dataset is structured with 100 slices per volume, where each slice corresponds to a different "layer" or depth in the 3D scan of the brain.
- **VOLUME_START_AT**: Set to 22, meaning the slicing will start from the 22nd slice of each volume.

By setting the start point to 22 and selecting 100 slices, this code effectively skips the first

22 slices and the last 22 slices (totaling 44 skipped slices out of 100). This selection helps to focus on the central part of each volume, which likely contains the most relevant anatomical information for the analysis.

3.4.3 Dataset Paths

- **TRAIN_DATASET_PATH**: Specifies the path to the BraTS 2020 Training Dataset.
- **VALIDATION_DATASET_PATH**: Specifies the path to the BraTS 2020 Validation Dataset.

3.4.4 Loading MRI Modalities

The **nibabel** library (**nib**) is used to load NIfTI files (file extension **.nii**), which store MRI data in 3D. Each image is loaded using **nib.load().get_fdata()** to read and extract the data as a NumPy array.

- **test_image_flair**: Loads the FLAIR image.
- **test_image_t1**: Loads the T1-weighted image.
- **test_image_t1ce**: Loads the T1-weighted contrast-enhanced image.
- **test_image_t2**: Loads the T2-weighted image.
- **test_mask**: Loads the segmentation mask for the patient.

Using these functions from NiLearn, we are able to plot the different modalities of the given data.

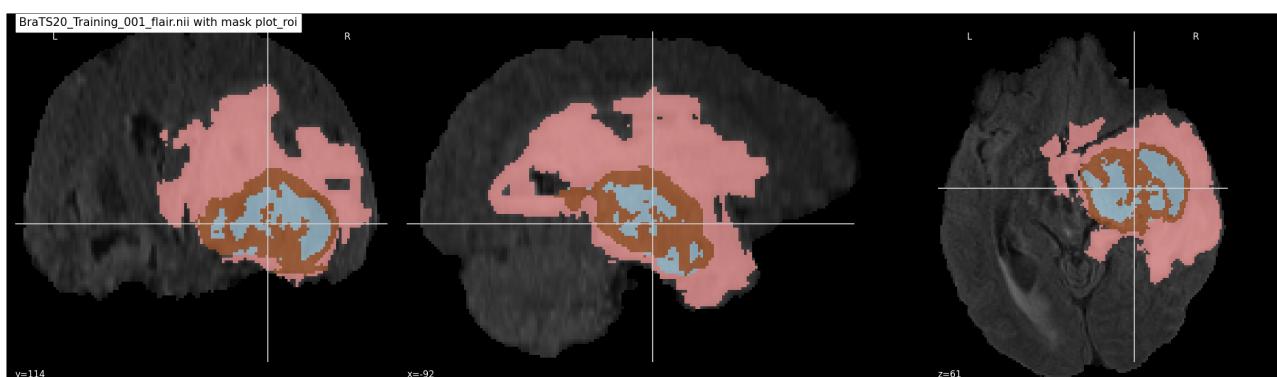
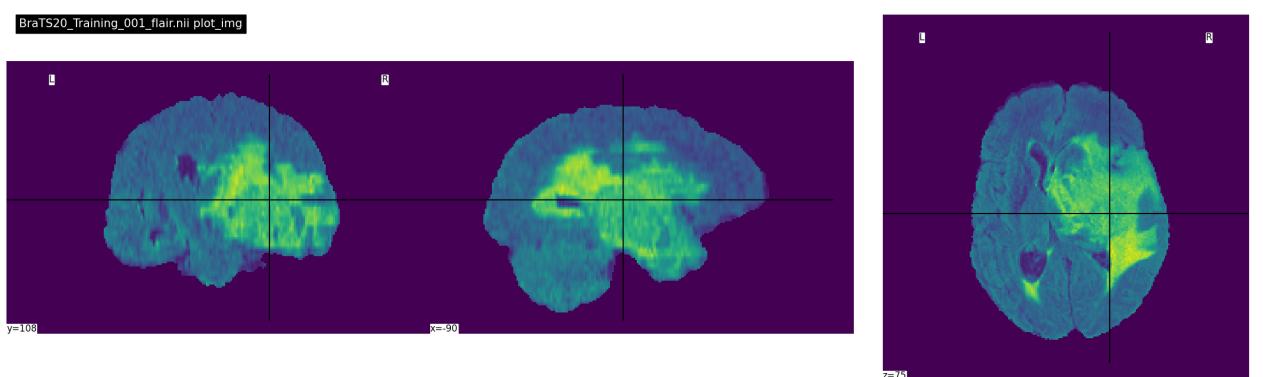
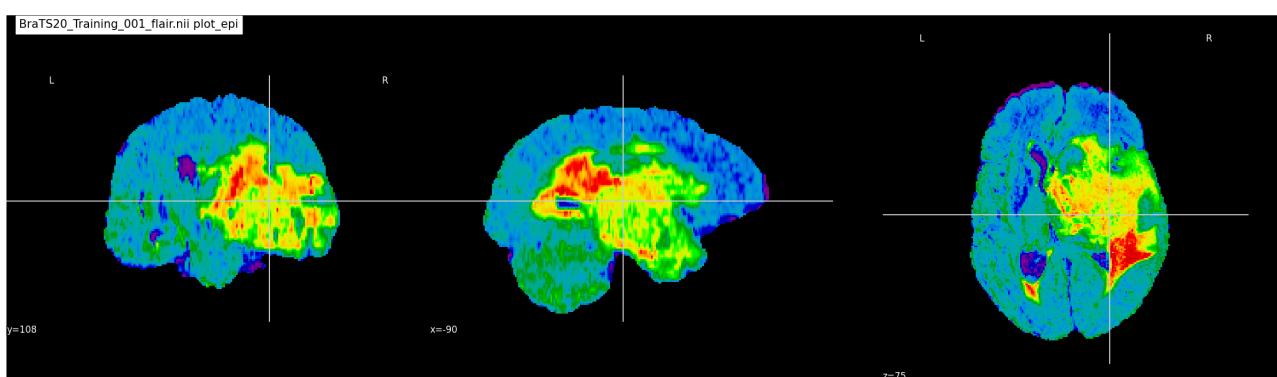
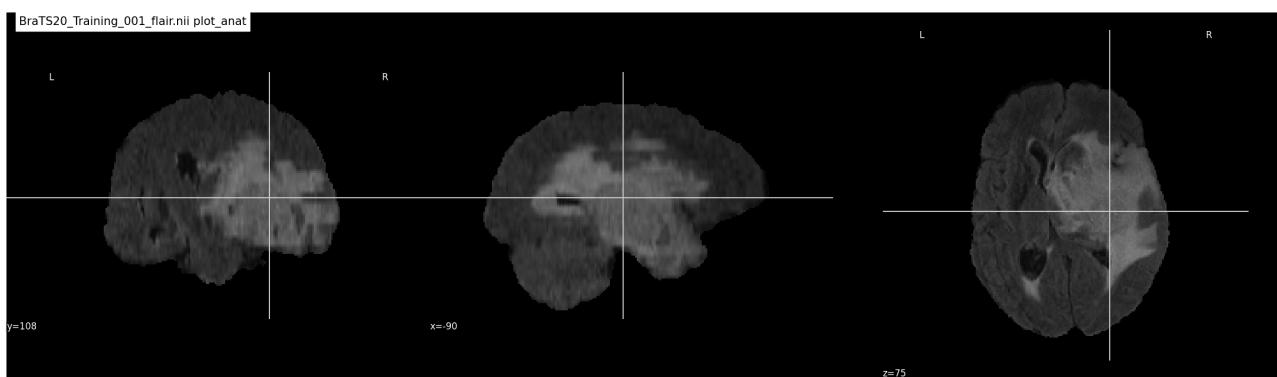


Figure 3.4: The data from BraTS dataset after completing all the pre-processing steps, ready to be loaded into a model of our choosing. Different visualisations have been shown with the help of the nilearn library.

4. U-Net Architecture

The U-Net architecture is a convolutional neural network (CNN) that is widely used for image segmentation, particularly in the field of medical imaging. Developed by Olaf Ronneberger et al. in 2015, U-Net is known for its ability to achieve precise pixel-level classification, even with a limited amount of labeled training data.

4.1 Architecture Overview

U-Net is an *encoder-decoder network*, consisting of a contracting path (encoder) that captures context and an expanding path (decoder) that enables precise localization. The name “U-Net” derives from the symmetric U-shape of the network, formed by the encoder and decoder paths.

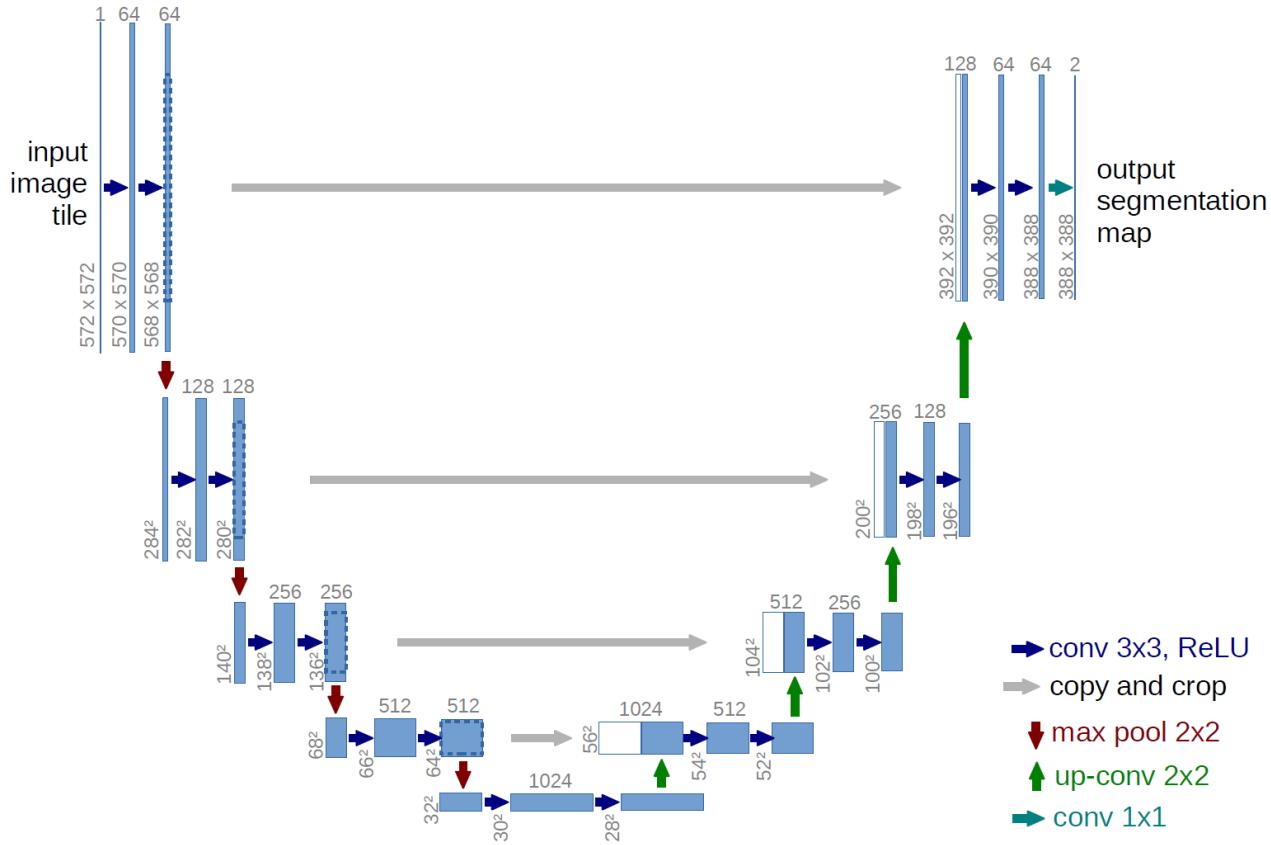


Figure 4.1: Schematic of the U-Net Architecture

4.1.1 Contracting Path (Encoder)

The encoder is composed of a series of convolutional layers followed by max-pooling operations:

- Each step in the encoder contains two 3×3 convolution layers (with ReLU activation) and a 2×2 max-pooling layer, which reduces the spatial dimensions by half.
- As the depth of the network increases, the number of feature channels doubles, allowing it to capture more complex and abstract representations.

4.1.2 Bottleneck

At the center of the U-Net architecture lies the *bottleneck*, which consists of convolutional layers that connect the lowest level of the encoder to the highest level of the decoder. The bottleneck captures the most abstract features before upsampling begins.

4.1.3 Expanding Path (Decoder)

The decoder path consists of upsampling layers that progressively increase the spatial resolution, followed by convolutional layers to reconstruct the segmented output:

- Each step in the decoder starts with an upsampling operation that doubles the spatial dimensions, followed by a 2×2 up-convolution (transpose convolution) to reduce the number of feature channels.
- The upsampled feature map is concatenated with the corresponding feature map from the encoder, allowing both high- and low-level information to be used during upsampling.

4.1.4 Skip Connections

A key feature of U-Net is the use of *skip connections*, where each layer in the encoder is connected to the corresponding layer in the decoder. This allows spatial information from the encoder to flow directly to the decoder, helping preserve fine-grained details in the segmentation map.

4.1.5 Final Layer

The final layer applies a 1×1 convolution to reduce the number of channels to the number of segmentation classes, followed by an appropriate activation function (e.g., *sigmoid* for binary segmentation or *softmax* for multi-class segmentation). This produces a pixel-wise classification for each pixel in the output map.

4.2 Loss Function

For segmentation tasks, U-Net typically uses a pixel-wise cross-entropy loss. In cases of imbalanced classes, a weighted cross-entropy or Dice coefficient loss can be used to improve performance.

4.3 Advantages

- **Efficient Training with Limited Data:** U-Net performs well even with limited labeled data due to data augmentation and its efficient design.
- **High Precision for Localization:** The skip connections and symmetric structure allow U-Net to maintain spatial accuracy, making it ideal for applications like medical imaging, where precise boundaries are important.
- **Flexible Input Sizes:** Without fully connected layers, U-Net can handle input images of varying sizes.

4.4 Applications

- **Medical Image Segmentation:** U-Net is widely used for segmenting organs, tumors, and other structures in medical images (e.g., MRI, CT).
- **Satellite Image Analysis:** U-Net can be applied to land cover classification and other tasks requiring precise segmentation.

- **Autonomous Driving:** U-Net is used to segment roads, pedestrians, and vehicles, aiding in scene understanding.

4.5 Extensions and Variants

- **3D U-Net:** An extension for volumetric data, primarily used in 3D medical imaging.
- **Attention U-Net:** Adds an attention mechanism to help the network focus on relevant features, improving performance on complex segmentation tasks.
- **ResU-Net:** Incorporates residual blocks for better gradient flow and improved performance on deeper networks.

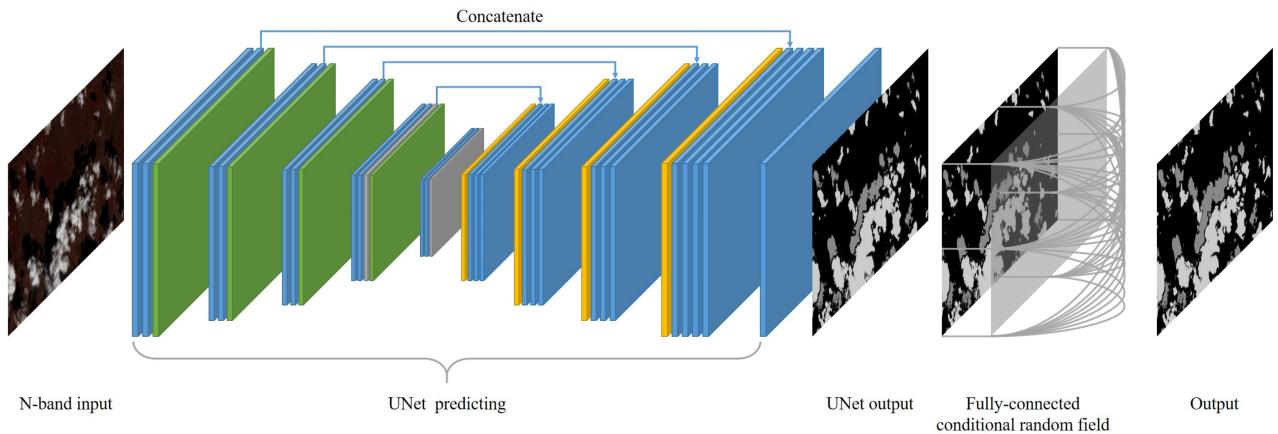


Figure 4.2: Detailed schematic of the U-net model.

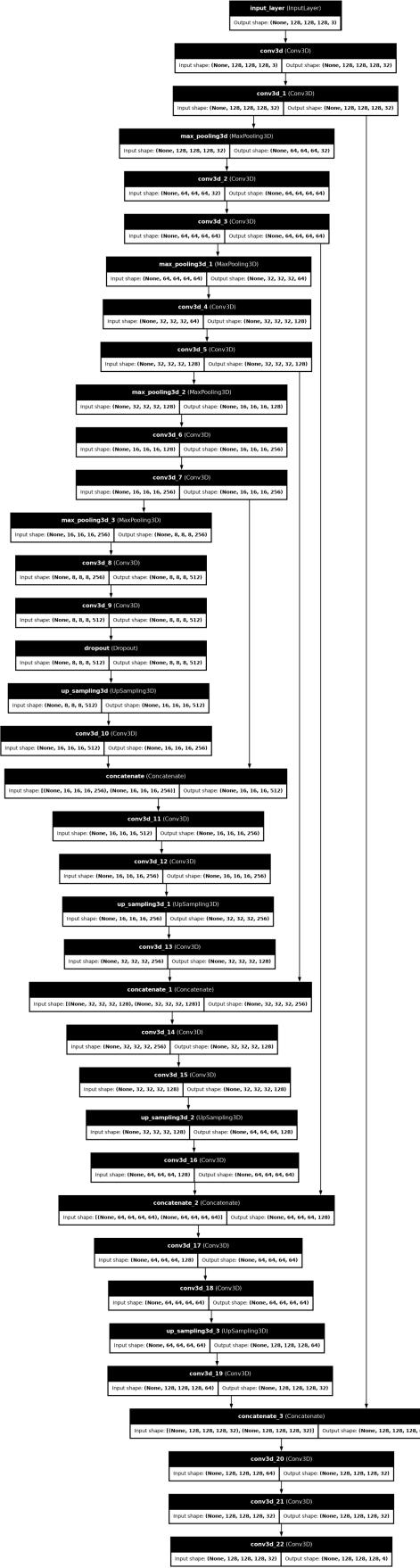


Figure 4.3: Model architecture generated by the keras drawing function.

5. Performance Metrics

5.1 Loss Function : Dice Coefficient

The Dice Coefficient is used as a measure of overlap between two samples, with values ranging from 0 to 1. A coefficient of 1 indicates perfect overlap, meaning the segmented output matches the ground truth completely. Originally developed for binary data, the Dice Coefficient can be calculated with the formula:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Figure 5.1: Formula for Dice Coefficient

$$|A \cap B| = \begin{bmatrix} 0.01 & 0.03 & 0.02 & 0.02 \\ 0.05 & 0.12 & 0.09 & 0.07 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{element-wise multiply}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix} \xrightarrow{\text{sum}} 7.41$$

prediction target

Figure 5.2: Matrix Representation of Dice Coefficient

5.2 Precision

Precision, also known as the Positive Predictive Value (PPV), measures the accuracy of the positive predictions made by the model. It tells us the proportion of predictions that were actually correct out of all the instances that the model classified as positive. Precision, also known as the Positive Predictive Value (PPV), is calculated as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

High precision indicates that, among all instances classified as positive, a large proportion were truly positive. In contexts like medical testing, a high precision ensures that when a test result is positive, it's highly likely to be accurate, which is crucial to minimize false alarms.

5.3 Sensitivity

Sensitivity, or Recall, measures the ability of the model to identify all relevant positive cases out of the actual positives in the data. It is a measure of how many true positive cases the model captures out of all the actual positive cases. Sensitivity, or Recall, measures the proportion of true positives that are correctly identified out of all actual positives:

$$\text{Sensitivity (Recall)} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

High sensitivity means that the model is effective at capturing actual positive cases, which is essential in cases where it's important not to miss true positives. In disease screening, for example, a high sensitivity ensures that most diseased patients are correctly identified.

5.4 Recall

Specificity, also known as the True Negative Rate (TNR), measures the ability of the model to correctly identify all negative cases. It indicates the proportion of true negatives among all the actual negatives. Specificity, also known as the True Negative Rate (TNR), is the proportion of true negatives correctly identified out of all actual negatives:

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}}$$

High specificity means the model is good at correctly identifying negative cases and avoiding false positives. In situations like fraud detection, where false positives can have operational costs, high specificity ensures that most legitimate cases are not mistakenly flagged as fraudulent.

6. Loading the Dataset

6.1 Data Generators

In deep learning tasks involving medical imaging, 3D segmentation models often work with vast amounts of data. Loading all of this data into memory can lead to significant memory issues, as the entire dataset may exceed the available memory capacity. To overcome this limitation, we can use **data generators**.

A data generator is a Python class that loads and processes batches of data on-the-fly, as needed during model training. Instead of loading the entire dataset at once, the generator loads a small batch of data, passes it through the model, and discards it before loading the next batch. This approach conserves memory and allows training on large datasets that would otherwise be infeasible to handle.

Data generators in Keras typically inherit from `tf.keras.utils.Sequence`, providing an efficient and thread-safe way to yield data batches. This method is commonly referred to as **data streaming** or **lazy loading**.

6.1.1 Benefits of Using Data Generators

Using data generators provides several advantages, particularly for handling large datasets in deep learning:

- **Reduced Memory Usage:** Only a small portion of data is loaded into memory at any time, helping to avoid memory overflow.
- **On-the-Fly Data Augmentation:** Generators can apply augmentations (e.g., rotations, flips) to each batch, enhancing data diversity without increasing storage requirements.
- **Scalability:** Generators enable models to be trained on datasets of virtually any size,

overcoming memory constraints.

- **Parallel Processing:** With `use_multiprocessing=True` and a specified number of workers, data loading can be parallelized, improving efficiency.

6.2 Data Split

To build robust machine learning models, it is essential to divide the dataset into separate groups for training, validation, and testing. This allows us to train the model, tune it, and evaluate its performance on unseen data. In this example, the `train_test_split` function from the `scikit-learn` library is used to achieve this division.

6.2.1 Splitting the Dataset

The dataset, which contains unique study IDs, is divided into three groups:

- **Training set (68%),**
- **Validation set (20%),** and
- **Testing set (12%).**

The process involves two steps to achieve these proportions:

Step 1: Creating `train_test_ids` and `val_ids`

First, the `train_test_split` function is used to split the initial dataset into two parts:

- `train_test_ids`: contains 80% of the total dataset.
- `val_ids`: contains 20% of the total dataset, used exclusively for validation.

This split allocates a portion of the dataset for validation, ensuring that 20% of the data is set aside and not used in model training or testing.

Step 2: Creating `train_ids` and `test_ids`

Next, the `train_test_split` function is applied again to the `train_test_ids` subset:

- `train_ids`: contains 85% of the `train_test_ids` data.
- `test_ids`: contains 15% of the `train_test_ids` data, used for testing.

Since `train_test_ids` represents 80% of the initial dataset, this split results in the following proportions for the original dataset:

- Approximately 68% for training ($0.85 \times 0.80 = 0.68$).
- 20% for validation (from `val_ids` in Step 1).
- Approximately 12% for testing ($0.15 \times 0.80 = 0.12$).

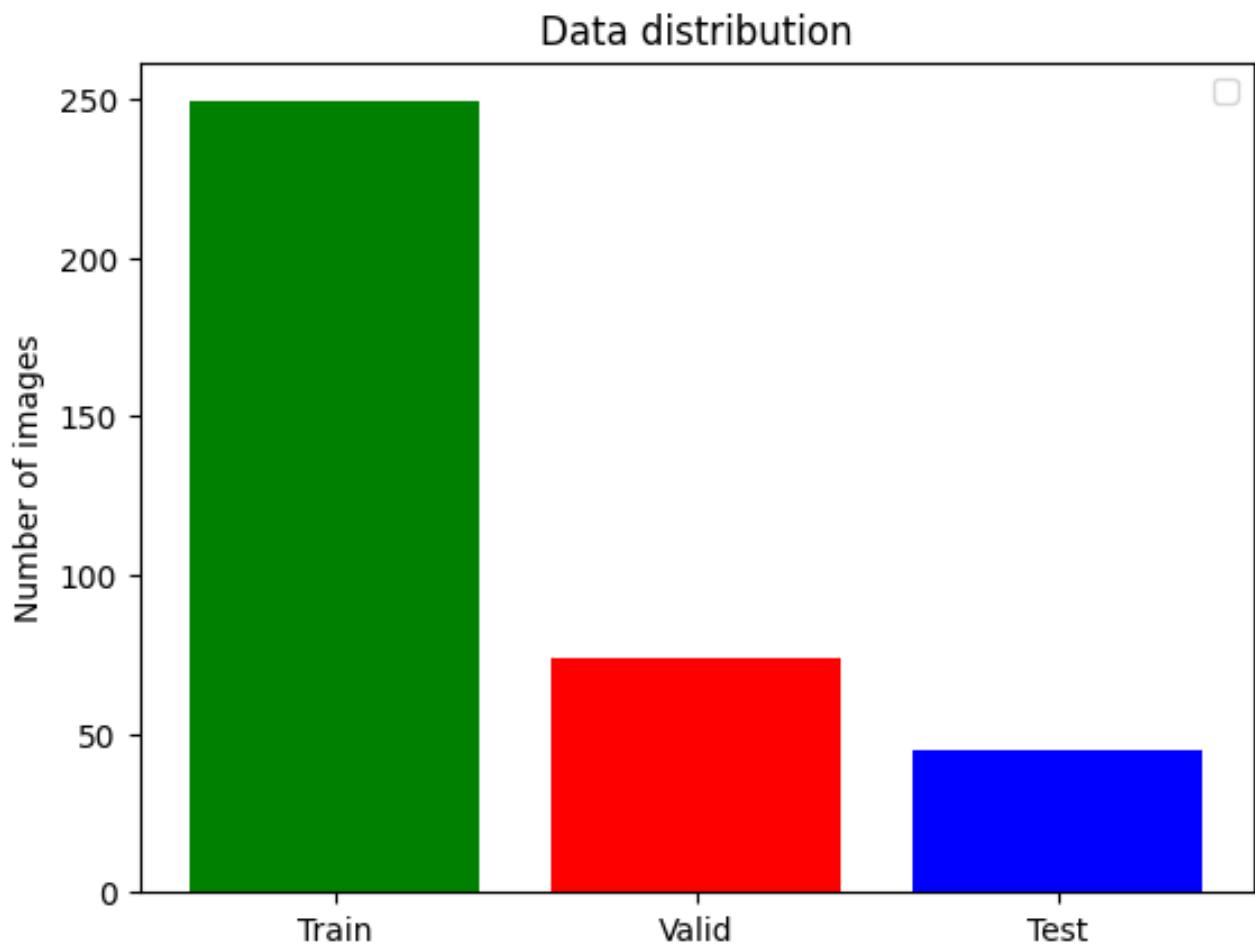


Figure 6.1: Visualisation of the split of data into training, test, validation sets.

7. Model Performance

The model is performing as follows :

- Accuracy = **99.2%**
- Loss = **0.024**
- Dice Coefficient = **0.62**
- IOU Score = **0.9**

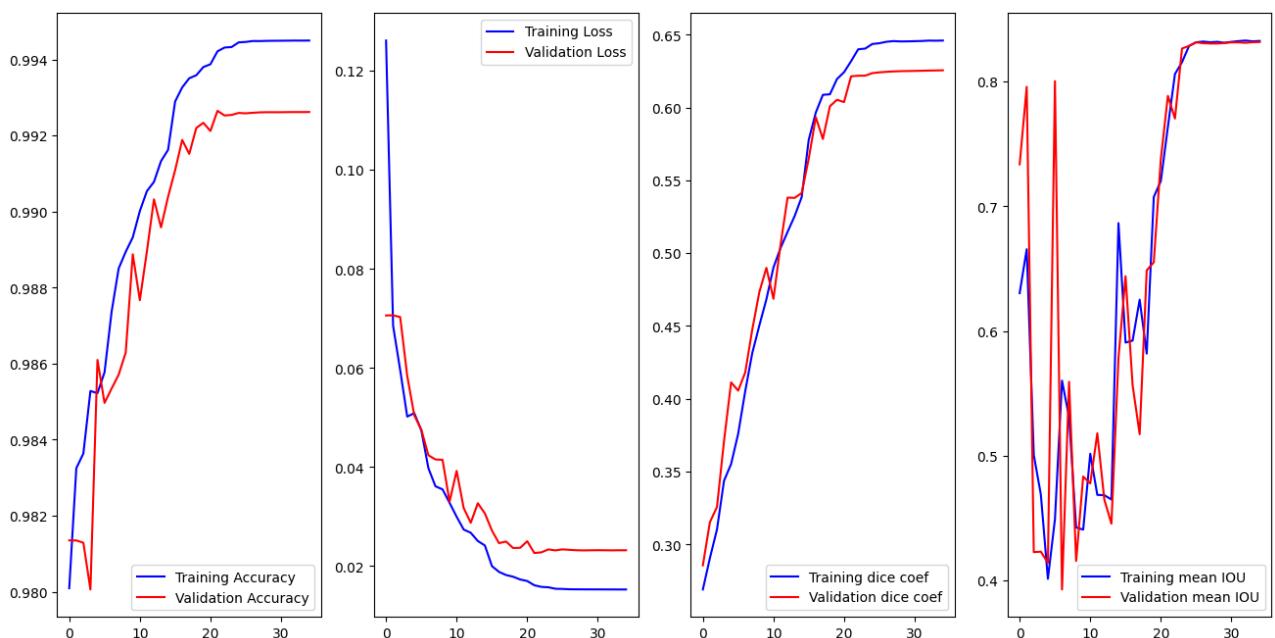


Figure 7.1: Performance Graphs of the model on all the metrics.

8. Predictions

This section contains some of the examples of the predictions that the segmentation model is able to perform :

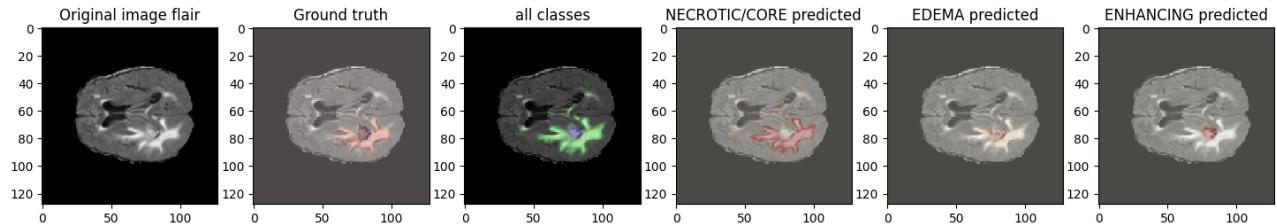


Figure 8.1: Example 1

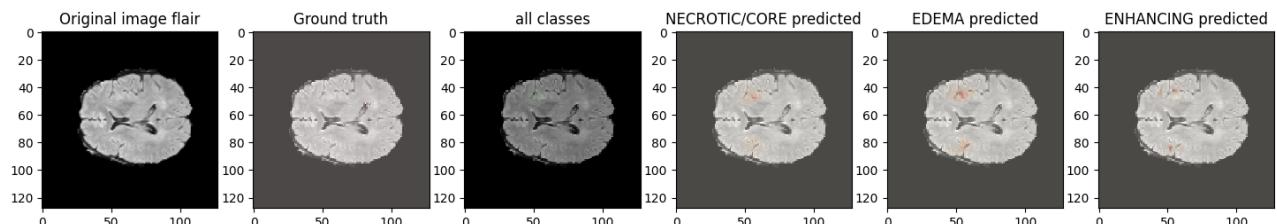


Figure 8.2: Example 2

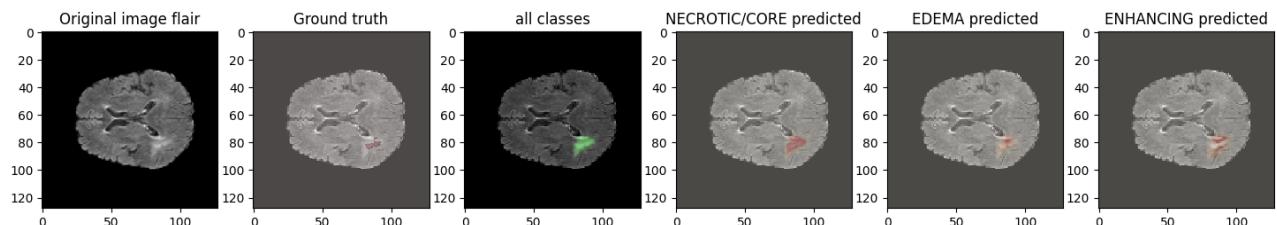


Figure 8.3: Example 3

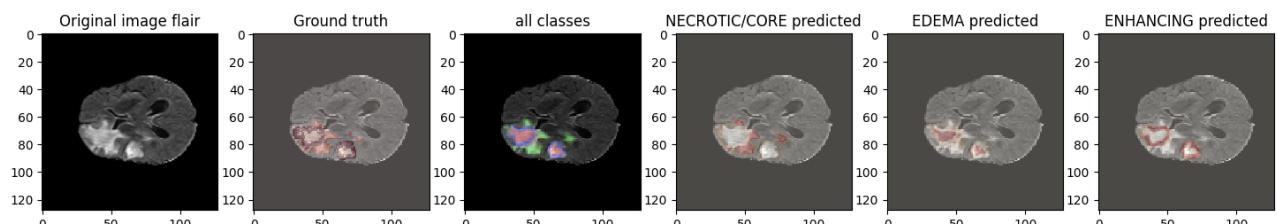


Figure 8.4: Example 4

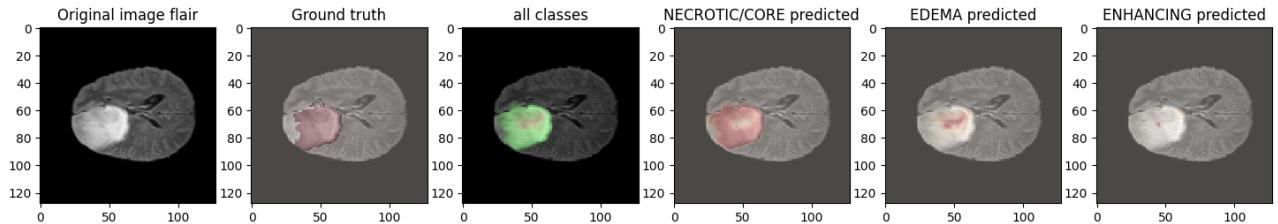


Figure 8.5: Example 5

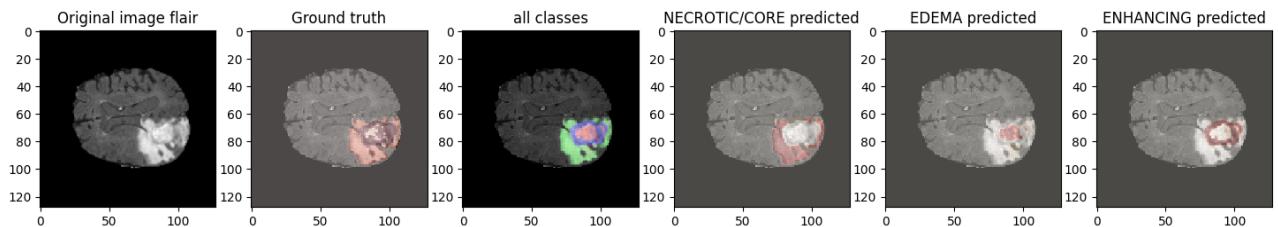


Figure 8.6: Example 6

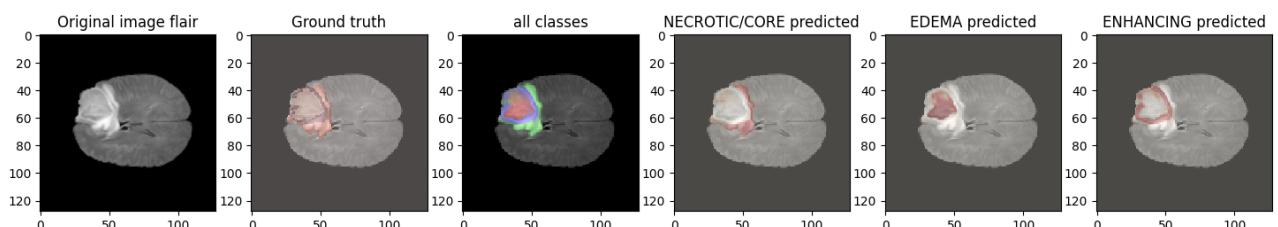


Figure 8.7: Example 7

8.1 Code and Resources

Link to Datasets :

BraTS19 - <https://www.kaggle.com/datasets/aryashah2k/brain-tumor-segmentation-brats-2019>

BraTS20 - <https://www.kaggle.com/datasets/awsaf49/brats20-dataset-training-validation>