Brain Tumor Segmentation using UNET architecture

Submitted By

Abduz Zami
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology

Supervised By

Tasmia Jannat
Lecturer
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology

Outline

1. Introduction
2. Objectives
3. Functions
4. Language and Tools with justification
5. Methodology
6. Scores

Introduction

1. BraTS 2020: Focuses on brain tumor segmentation in MRI scans
2. Evaluation of methods for glioma segmentation, patient survival prediction, progression status assessment, and segmentation uncertainty estimation
3. Data includes multi-institutional MRI scans with manual tumor segmentations and clinical information
4. Scans cover native, post-contrast, T2-weighted, and T2-FLAIR volumes
5. Annotations include enhancing tumor, peritumoral edema, and tumor core
6. Job: Automation of segmentation process using U-NET

Objectives

1. Creating a UNET architecture that can generate segmented 3D mask images containing 3 types of annotations i.e. enhancing tumor, peritumoral edema, and tumor core.
2. Enhancing Dice Coefficient, IoU score, precision, recall, and F1-score.
3. Surpass related works on it.

Functions

From any 3D MRI of the brain, this trained model can generate segmentation containing 3 types of annotations i.e. **enhancing tumor**, **peritumoral edema**, and **tumor core**.

Language & Tools with justification

1. **Python**: Programming language used for the project.
2. **Nilearn**: Library utilized for handling and visualizing neuroimaging data, especially NIfTI files.
3. **Segmentation Models 3D**: Libraries accessed for pre-built architectures and utilities tailored for 3D medical image segmentation and classification tasks.
4. **Visualkeras**: Tool employed for visualizing the architecture of neural network models.
5. **NumPy**, **scikit-learn**, **Matplotlib**: Standard libraries used for data preprocessing, model evaluation, and result visualization.

Methodology

I will describe the following topics under methodology in the next slides:

1. **Dataset**
2. **Preprocessing**
3. **Model Creation**
4. **Experimental Set Up**

Methodology (Cont'd)

Dataset

Dataset name: **BraTS2020**

Dataset Size: **369** items for training and **125** items for validating

(Only worked with the training part of the dataset for both training and validation)

Each item has **5 .nii** type files.

They contain **native**, **post-contrast**, **T2-weighted**, **T2-FLAIR** volumes and one **segmentation** file.

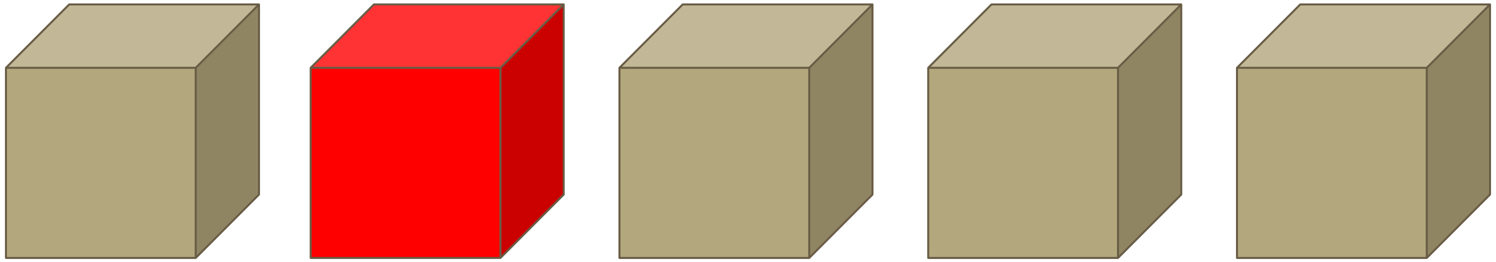
Segmentation file contains **3** types of annotations: **enhancing tumor**, **peritumoral edema**, and **tumor core**.

Each .nii file has a shape of **(240, 240, 155)**.

Methodology (Cont'd)

Preprocessing

5 images in a single folder

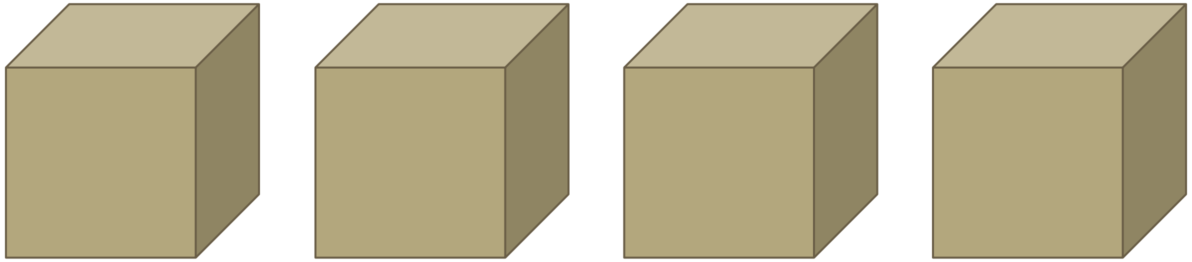


T1	T2	Flair	T1CE	Seg
(240, 240, 155)	(240, 240, 155)	(240, 240, 155)	(240, 240, 155)	(240, 240, 155)

Methodology (Cont'd)

Preprocessing

Removed T2 image

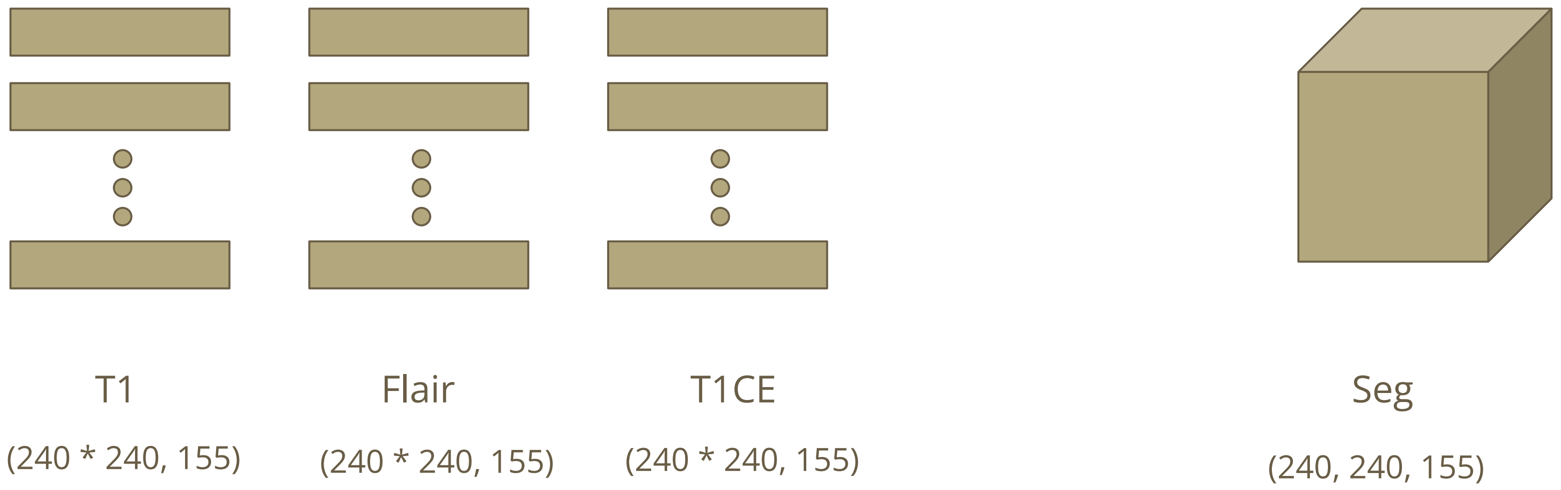


T1	Flair	T1CE	Seg
(240, 240, 155)	(240, 240, 155)	(240, 240, 155)	(240, 240, 155)

Methodology (Cont’d)

Preprocessing

T1, Flair, T1CE were flatten on the first two axis. And MinMaxScalling were applied to them.



T1

(240 * 240, 155)

Flair

(240 * 240, 155)

T1CE

(240 * 240, 155)

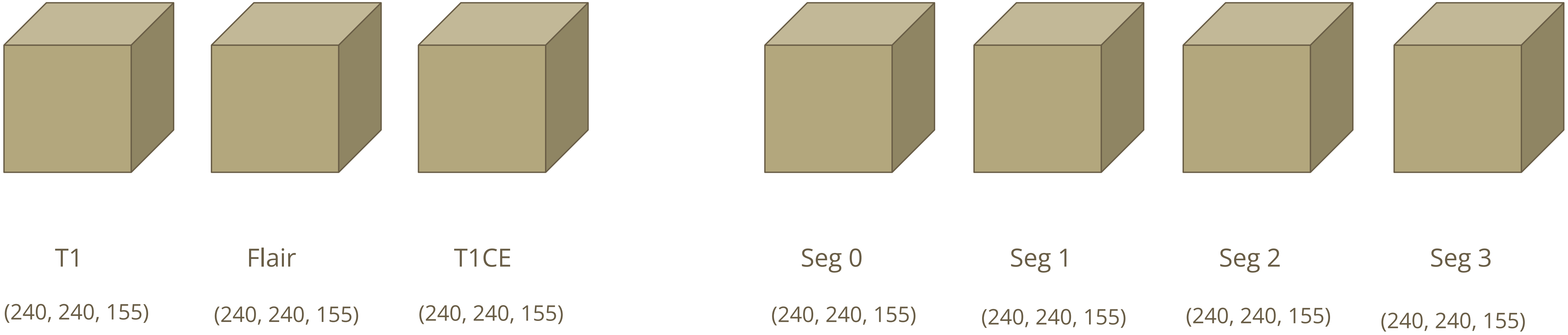
Seg

(240, 240, 155)

Methodology (Cont’d)

Preprocessing

1. T1, Flair, T1CE were brought back to original shape.
2. One Hot Encoding were applied on Segmentation Image. Segmentation image had four unique pixel values 0 (Nothing), 1 (Non-enhancing tumor core), 2 (Edema), 3 (Not Specified), 4 (Enhancing tumor).
3. Label 3 was replaced by label 4.



T1 (240, 240, 155)

Flair (240, 240, 155)

T1CE (240, 240, 155)

Seg 0 (240, 240, 155)

Seg 1 (240, 240, 155)

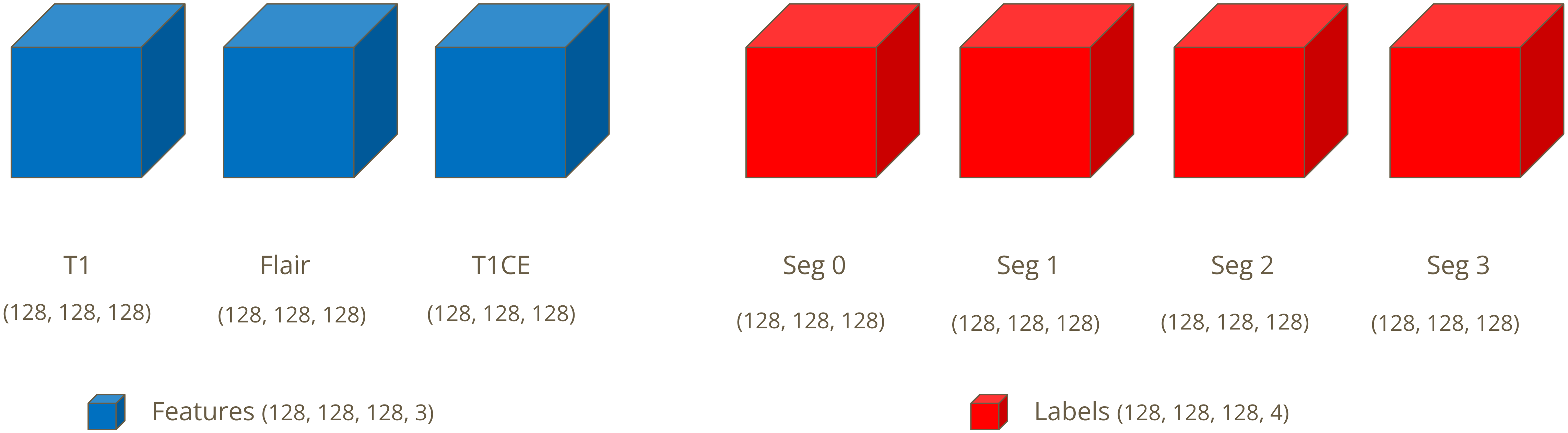
Seg 2 (240, 240, 155)

Seg 3 (240, 240, 155)

Methodology (Cont’d)

Preprocessing

T1, Flair, T1CE and segmentations were cropped to (128, 128, 128)



T1 (128, 128, 128)

Flair (128, 128, 128)

T1CE (128, 128, 128)

Seg 0 (128, 128, 128)

Seg 1 (128, 128, 128)

Seg 2 (128, 128, 128)

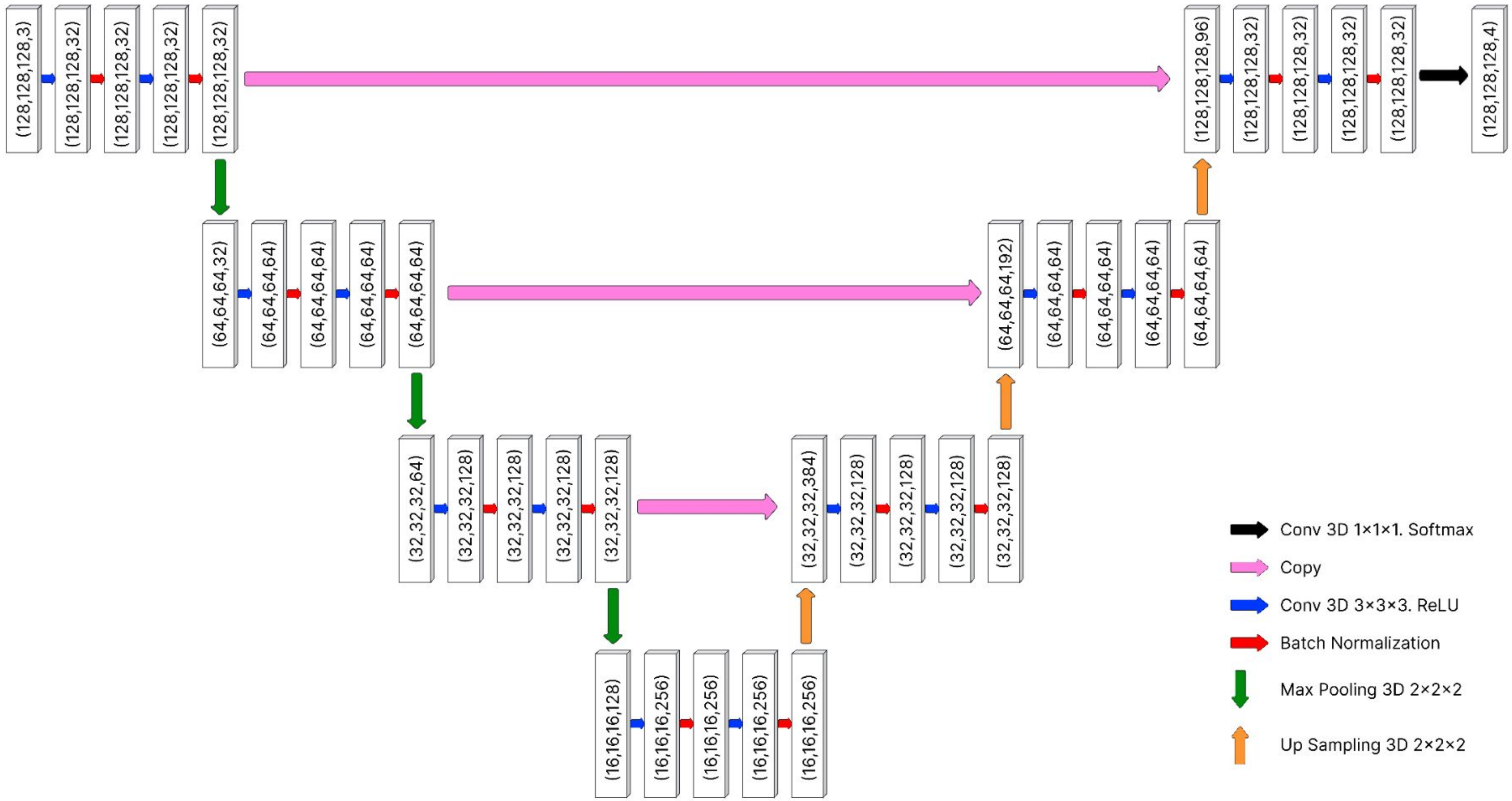
Seg 3 (128, 128, 128)

🟦 Features (128, 128, 128, 3)

🟥 Labels (128, 128, 128, 4)

Methodology (Cont’d)

Model Creation



Methodology (Cont’d)

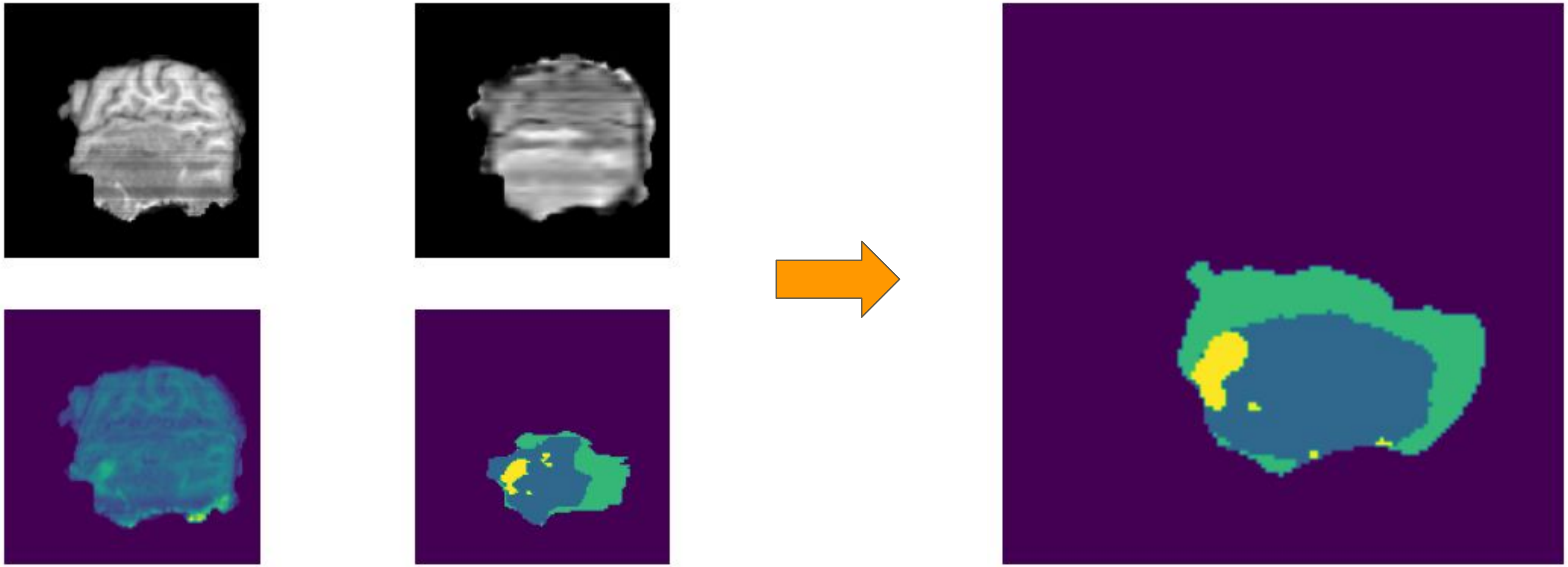
Environmental Set Up

1. I split the Training Folder into train(80%) and test(20%).
2. I further split the train set into train(80%) and validation(20%).
3. I trained the model for 50 epochs with a batch size of 1.
4. Loss function was totalloss=diceloss+(1*focalloss).
5. Learning rate was 0.001.
6. Optimizer was Adam.
7. Early Stopping was applied with patience 5 on validation loss.
8. For evaluation metrics I used Dice Coefficient, Intersection over Union,

Scores

Label	Dice-Coefficient (%)	IoU (%)
0	98.97	98.05
1	55.20	44.59
2	62.26	49.98
3	62.17	51.94

Scores (Cont’d)



Thank You