**This is documentation for Brain Tumor Segmentation Project**

***Names of all files used in this project***

1. Config.py
2. Dataset.py
3. Feature\_extraction.py
4. Inference.py
5. Loss.py
6. Main.py
7. Process\_dataset.py
8. Trainer.py
9. Unet\_model.py
10. Visualize\_results.py
11. Segmenation\_notebook.ipynb

**Config.py**

In this file we created Config class in which we set all global variables and parameters.

**Dataset.py**

In this file we created two classes:

* + **DatasetCreator**:
    - This class is responsible for splitting our dataset (images and masks) into test and train sets
  + **TumorDataset**:
    - This class is responsible for creating datasets for testing and training from test and train sets

**Feature\_extraction.py**

This file has one function “extract\_features” which takes image and mask, and finds features

Perimeter

* Area
* Radius
* Diameter
* Circularity
* Eccentricity
* texture\_features

**Inference.py**

This file consists of one function “inference” which loads images from inference folder and does inference on all images, saves results in combined\_image\_mask folder.

**Loss.py**

This file has two class:

* + **BCEDiceLoss**:
    - This class is used to calculate loss binary\_cross\_entropy + dice\_loss, we used it in our training.
  + **DiceLoss**:
    - This class is used to calculate dice\_loss, we used it in validation.

**Main.py**

This file creates two folders in which model weights are to be stored during training process, and trains segmentation model.

**Process\_dataset.py**

This file has three functions

* + **Makedir**:
    - Creates directory (folder)
  + **Unzip**:
    - Unzips zip file(s)
  + **Convert\_matfiles\_to\_images**:
    - Extracts images and masks from matfiles

**Trainer.py**

This file has a class of Trainer

* **constrcutor** provides model with best weights, loss funtions, and device
* **setup\_training\_env()**, the setup for the training prepares eg dataset, dataloaders and optimizer
* **train\_fn(),** trains the model and saves best weights and optimizer
* **save\_results\_to\_csv()**, saves history in csv file

**Unet\_model.py**

This file has four helper classes

* + DoubleConv
  + Down
  + Up
  + OutConv

These classes are used in Unet class to make Unet segmentation model

**Visualize\_results.py**

This file has a function plot\_loss which is used to visualize training matrices

**Segmenation\_notebook.ipynb**

Flow of running files + full pipeline for inference is available in this notebook



