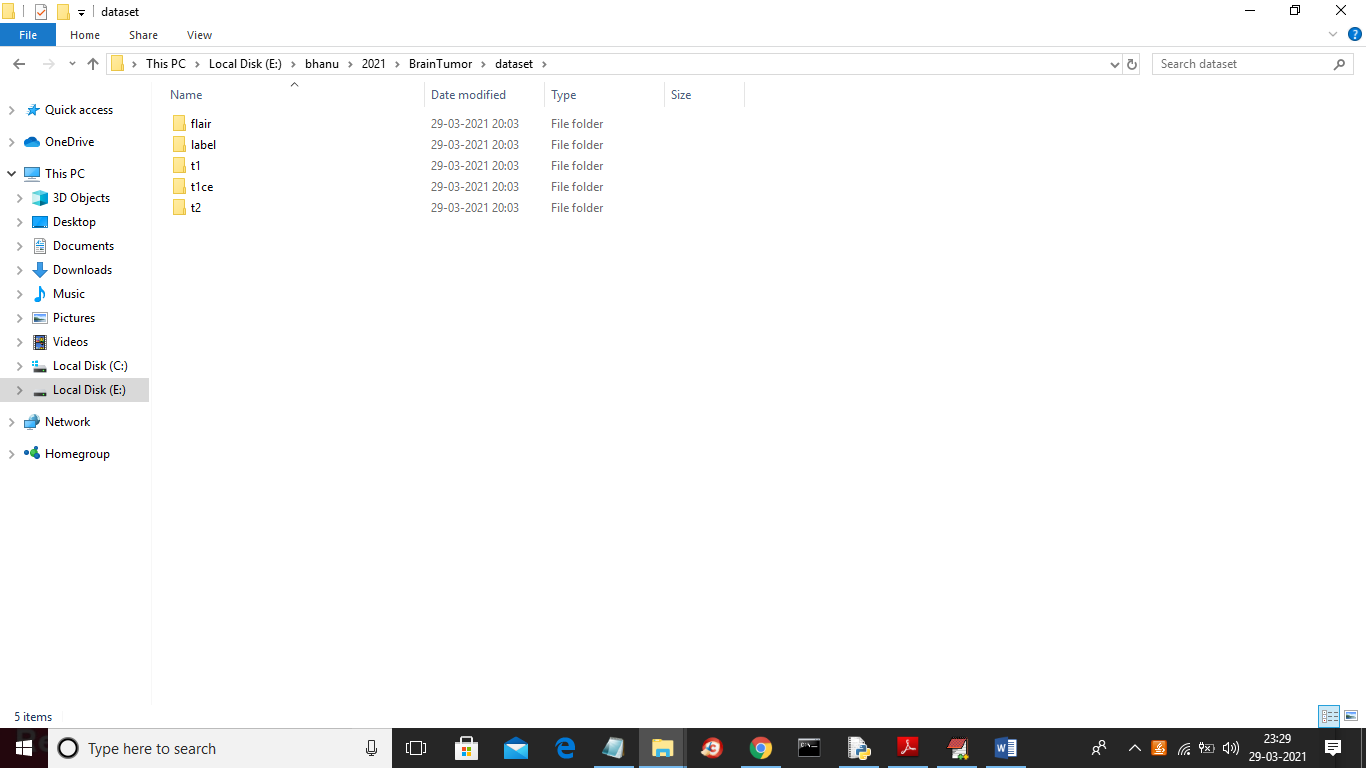
Brain Tumour Image Segmentation Using Deep Networks

To automate brain tumour segmentation process author is combining both 3D CNN and UNET algorithms as deep learning is gaining popularity in efficient semantic segmentation of medical images. To further enhance segmentation process author is using combination or ensemble of two deep learning algorithms called CNN and UNET. Both algorithms trained separately on BRATS brain tumour dataset and then predicted output of both algorithms will be merge or map to generate final segmentation and the output generated is giving high dice score after mapping both algorithms segmentation and then predicting final segmented output. Dice score refers to correctly mapping of segmented parts in the image.

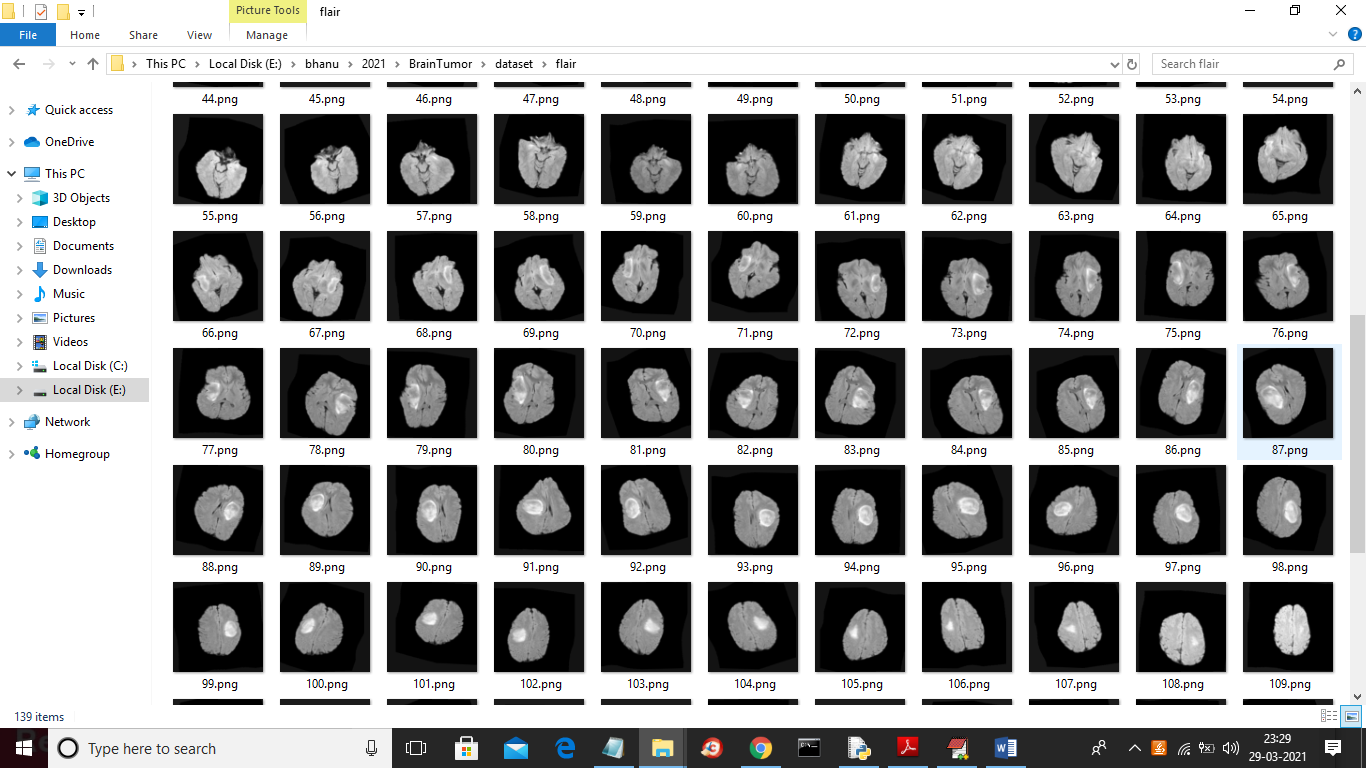
The task is to develop an automated brain tumour segmentation method, for successful delineation of tumours into intra-tumoural classes with improved efficiency and accuracy in comparison to existing methods.

To implement this project we are using 4 different images and this images are called as FLAIR, T1, T2 and T1CE and the label segmented image. The multi-institutional dataset, acquired from 19 different contributors, contains multimodal MRI scans of each patient, namely T1, T1 contrast-enhanced (T1ce), T2-weighted (T2), and Fluid Attenuated Inversion Recovery (FLAIR), from which the tumoural sub regions are segmented. The data is processed to overcome discrepancies such that they are skull-stripped.

BRATS dataset images are saved inside dataset folder and in below screen you can see dataset content

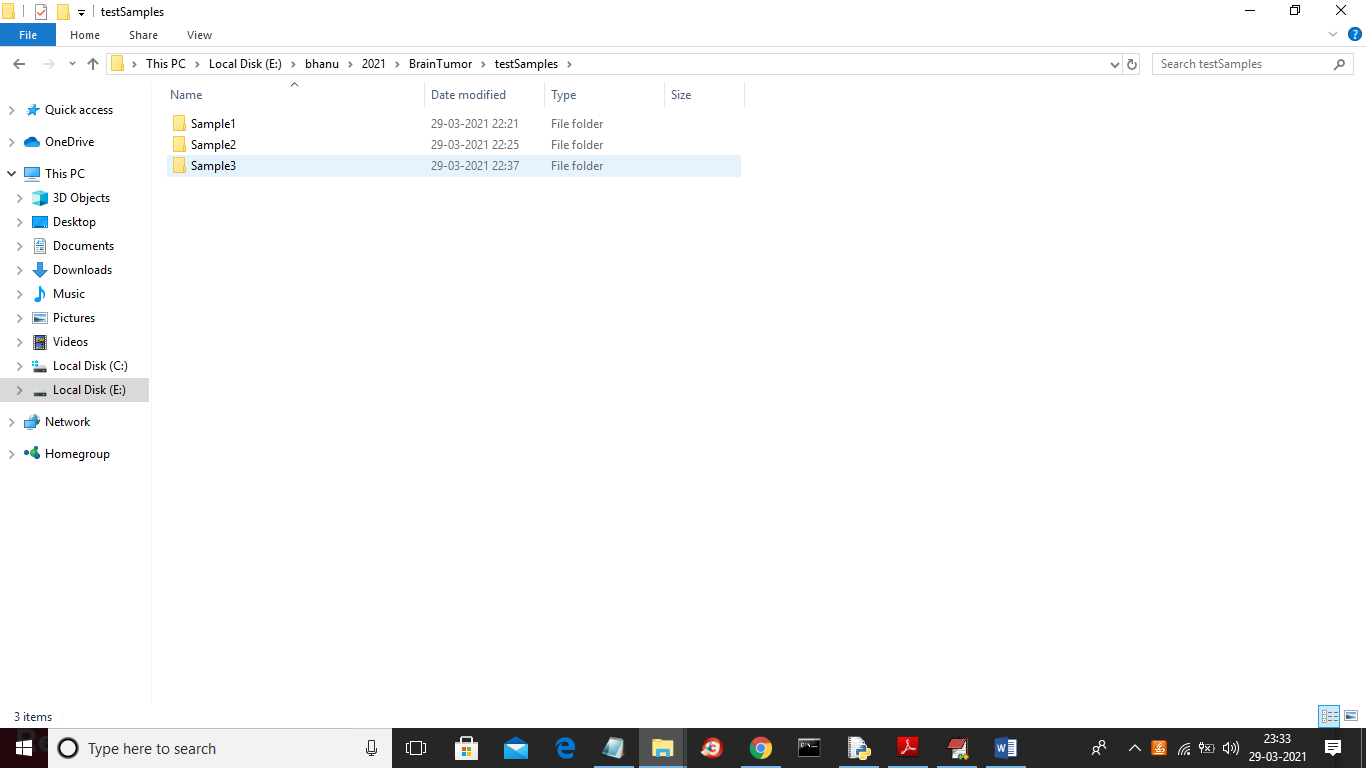


In above screen we have different format image and you can go inside any folder to see images

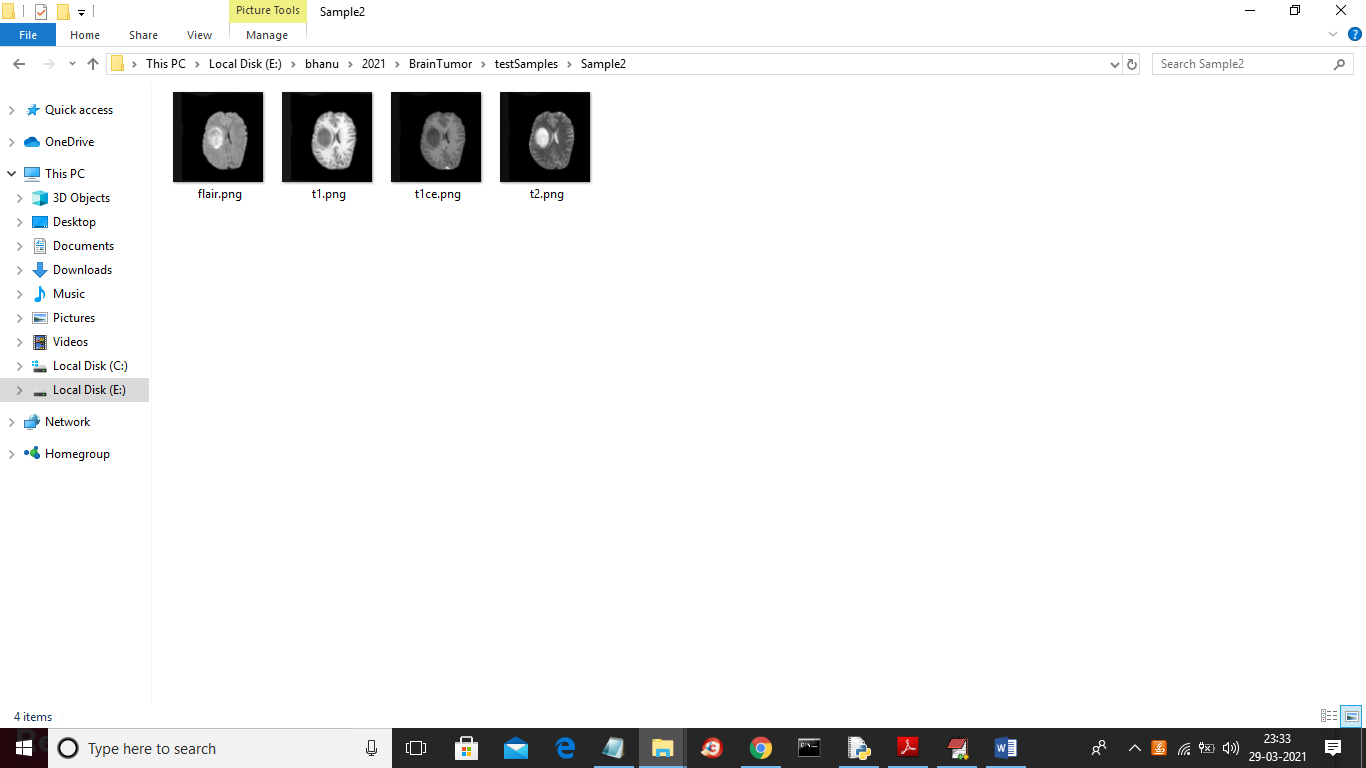


Above dataset is used to train CNN and UNET model

After building UNET and CNN model we will upload test images from ‘testSamples’ folder and then UNET model will give us segmented image. Below screen shots showing testSamples image



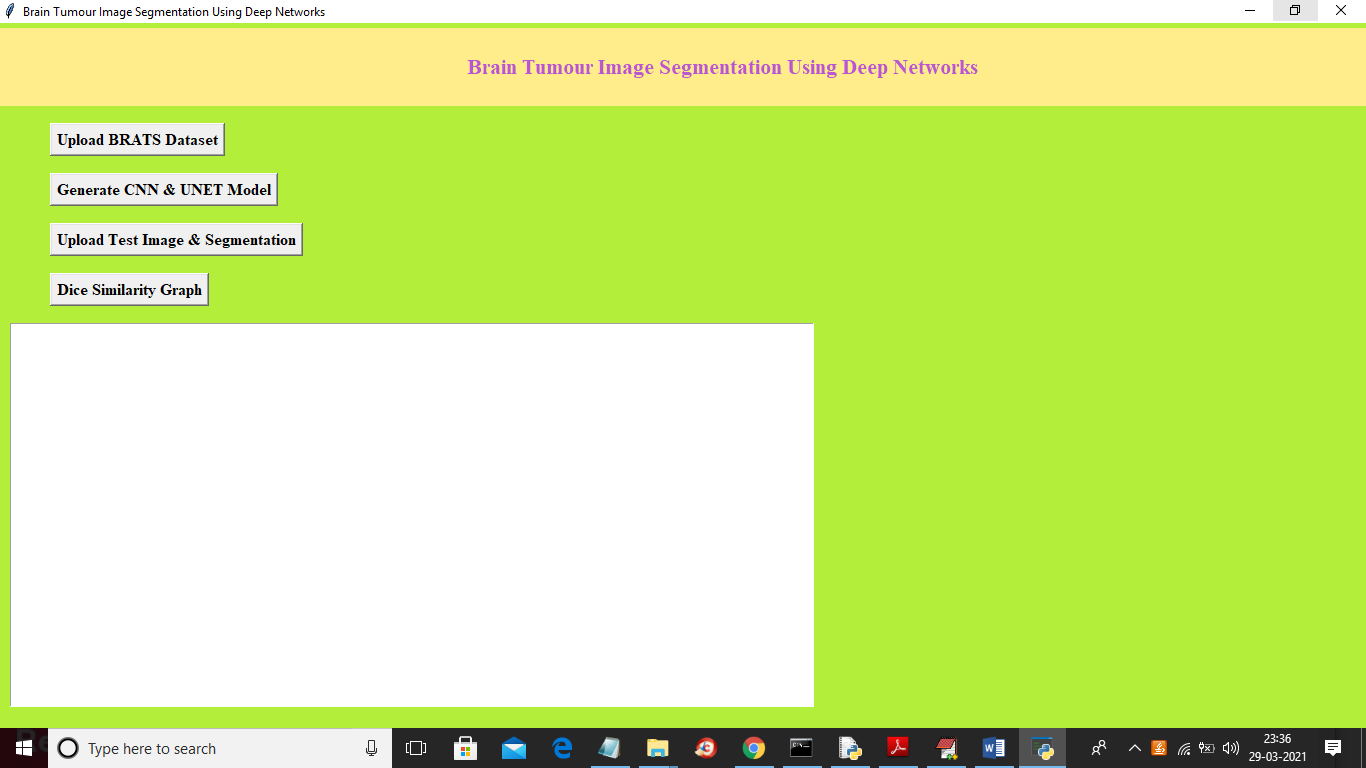
In above screen we have 3 samples images and now go inside any folder to get below images



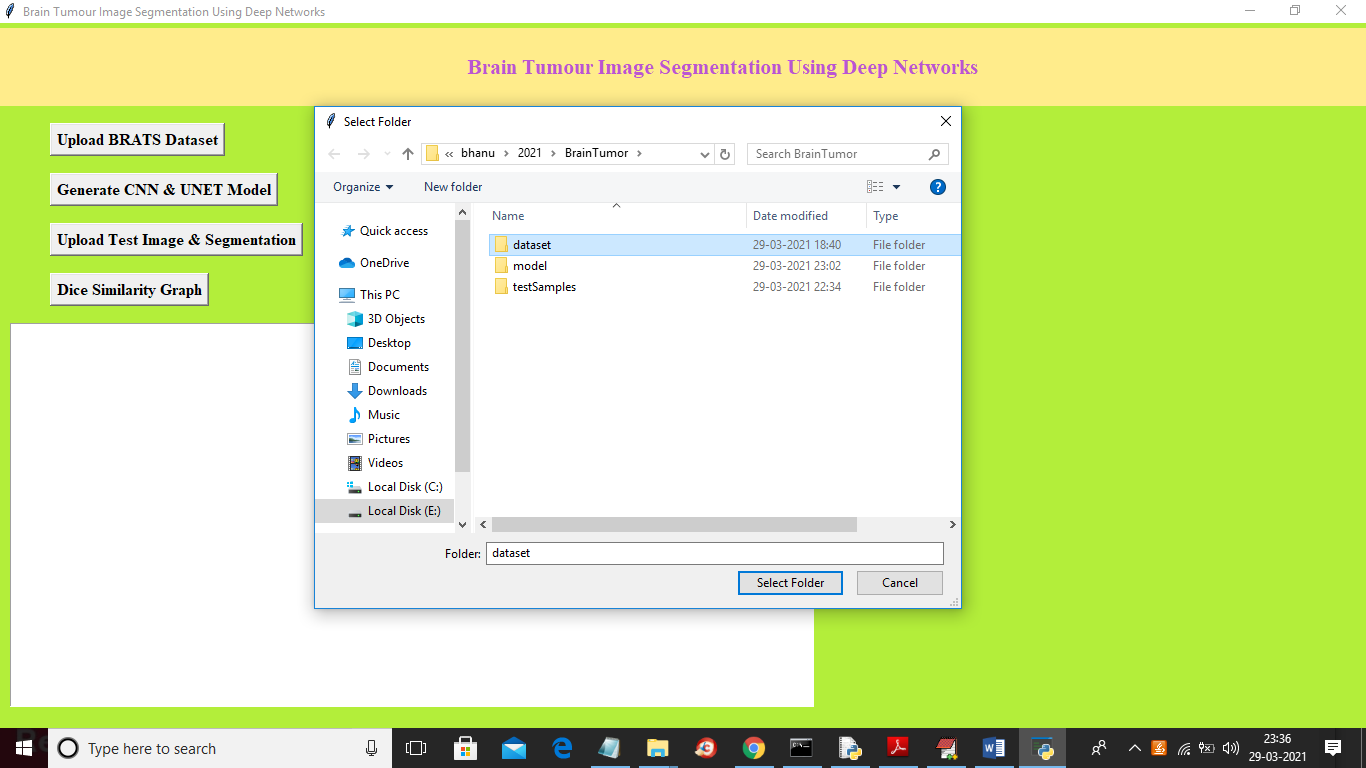
In above screen we have FLAIR, T1, TICE and T2 images but we don’t have segmented label image and after applying model on above images then we will get segmented label image

SCREEN SHOTS

To run project double click on ‘run.bat’ file to get below screen



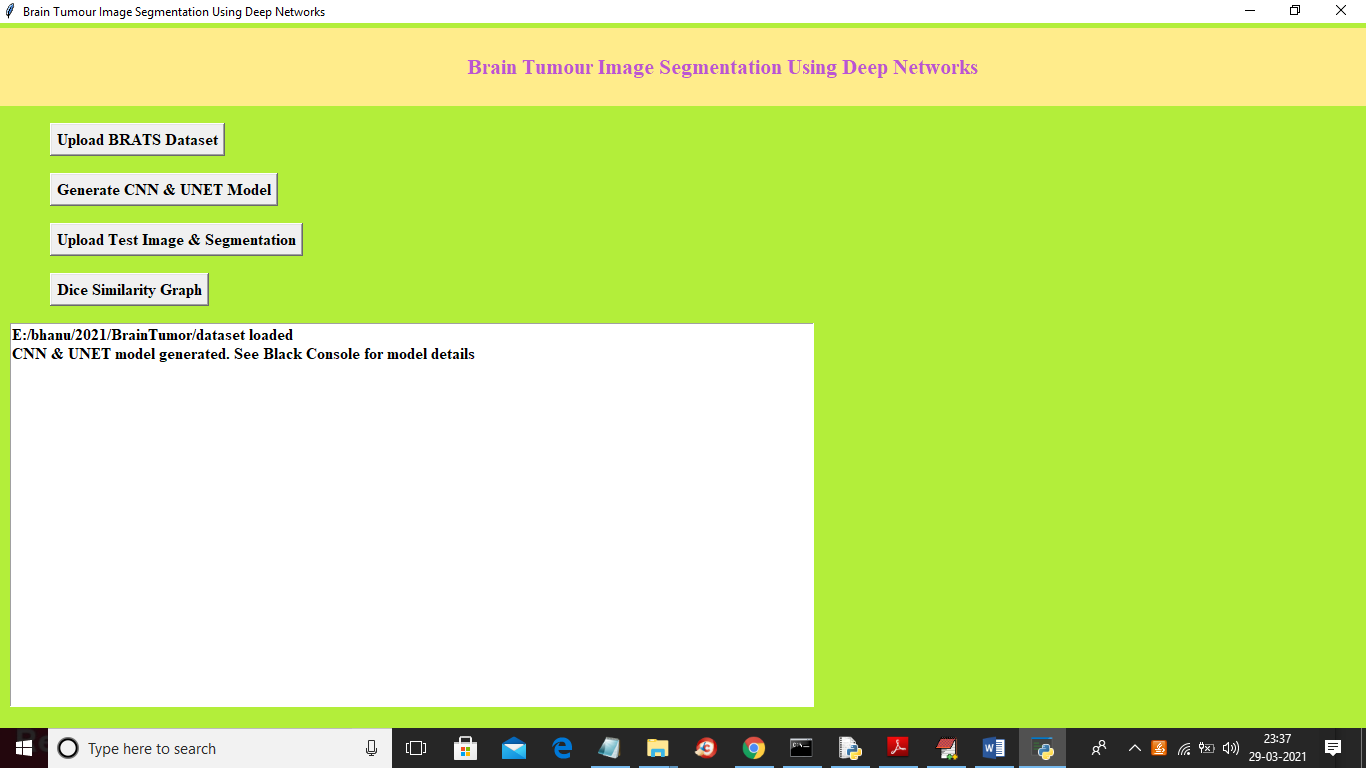
In above screen click on ‘Upload BRATS Dataset’ button to upload dataset



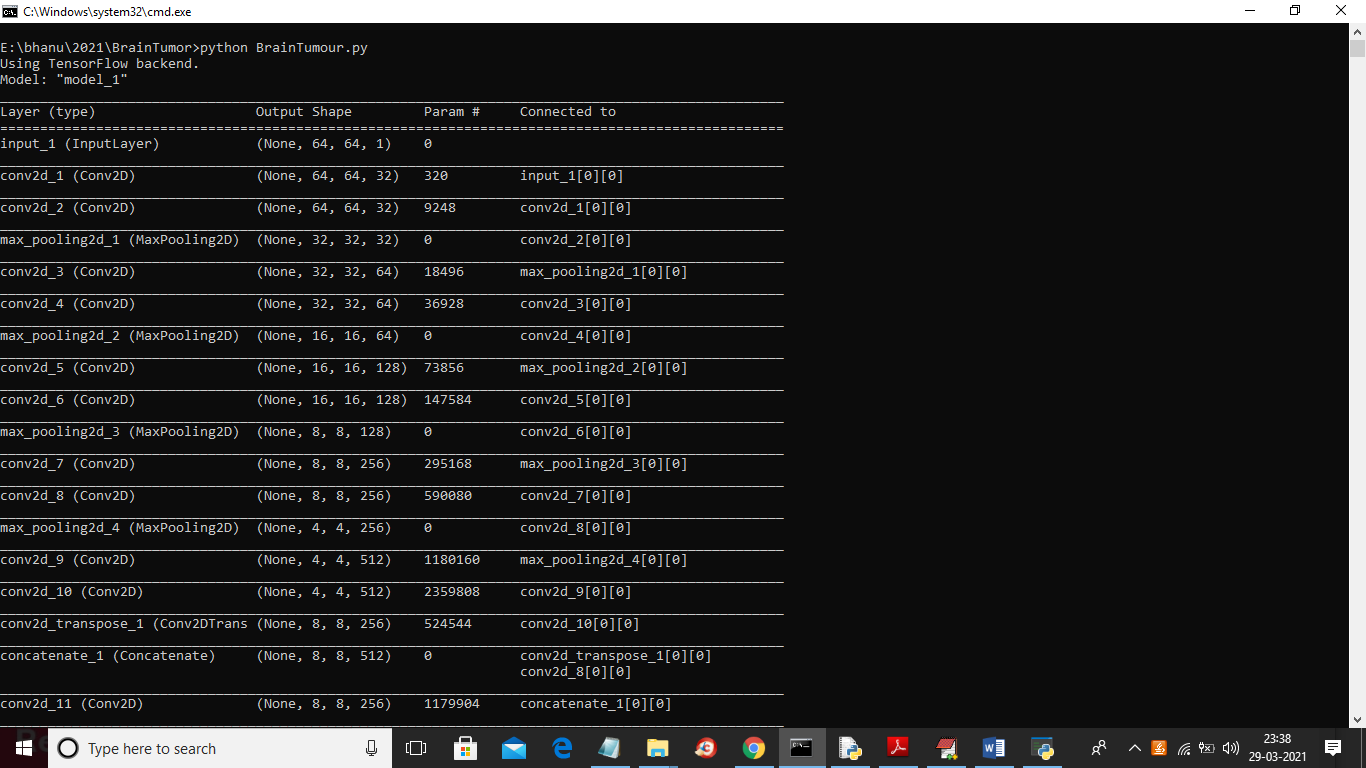
In above screen selecting and uploading ‘dataset’ folder and then click on ‘Select Folder’ button to load dataset and to get below screen



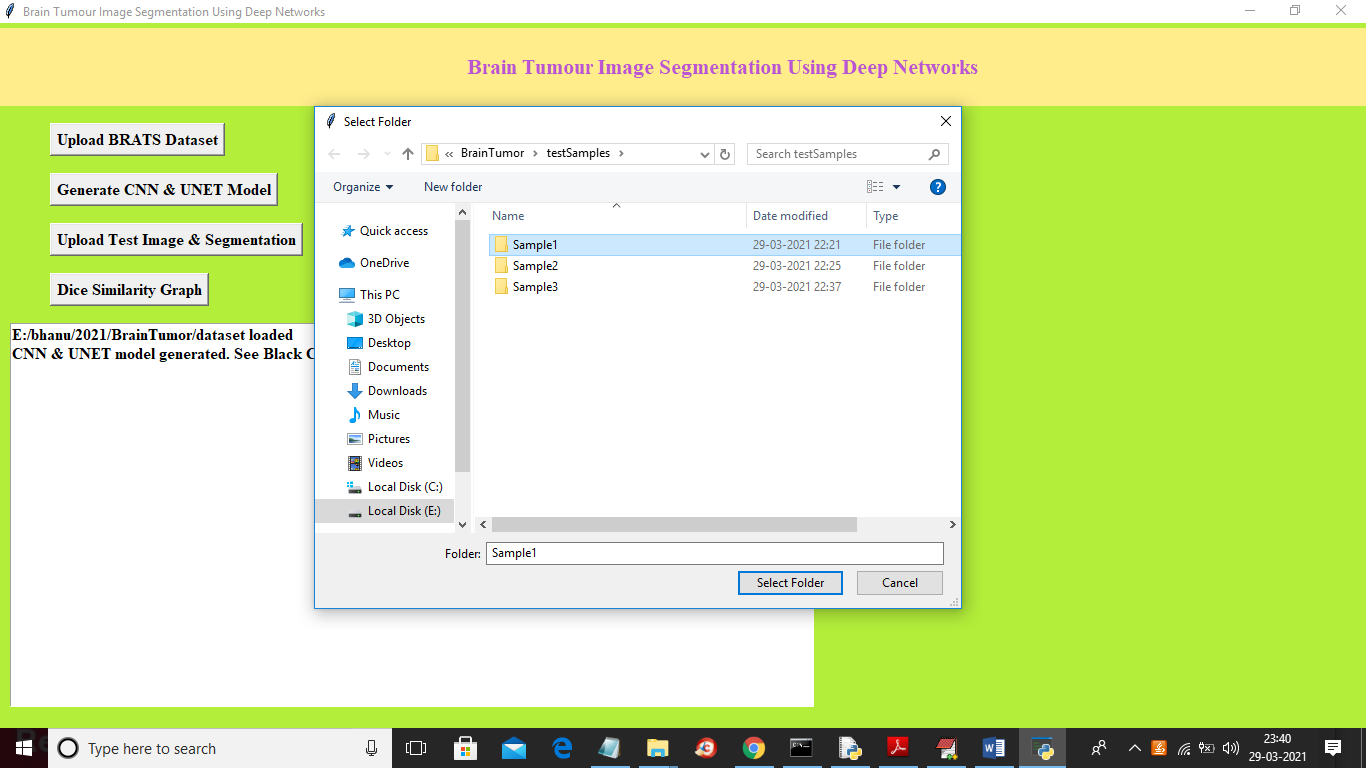
In above screen dataset loaded and now click on ‘Generate CNN & UNET Model’ button to generate models and to get below screen



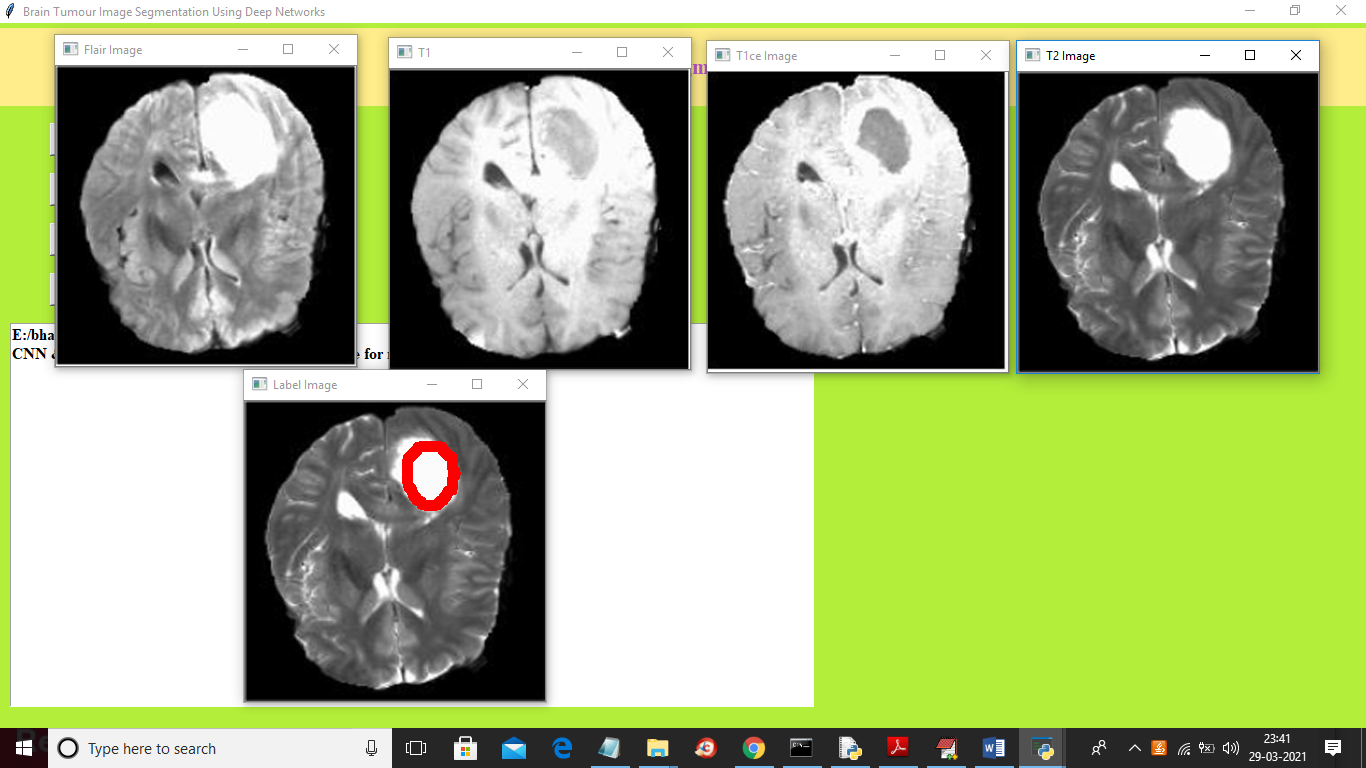
In above screen we can see both models are generated and we can see below black console to see CNN and UNET layer details



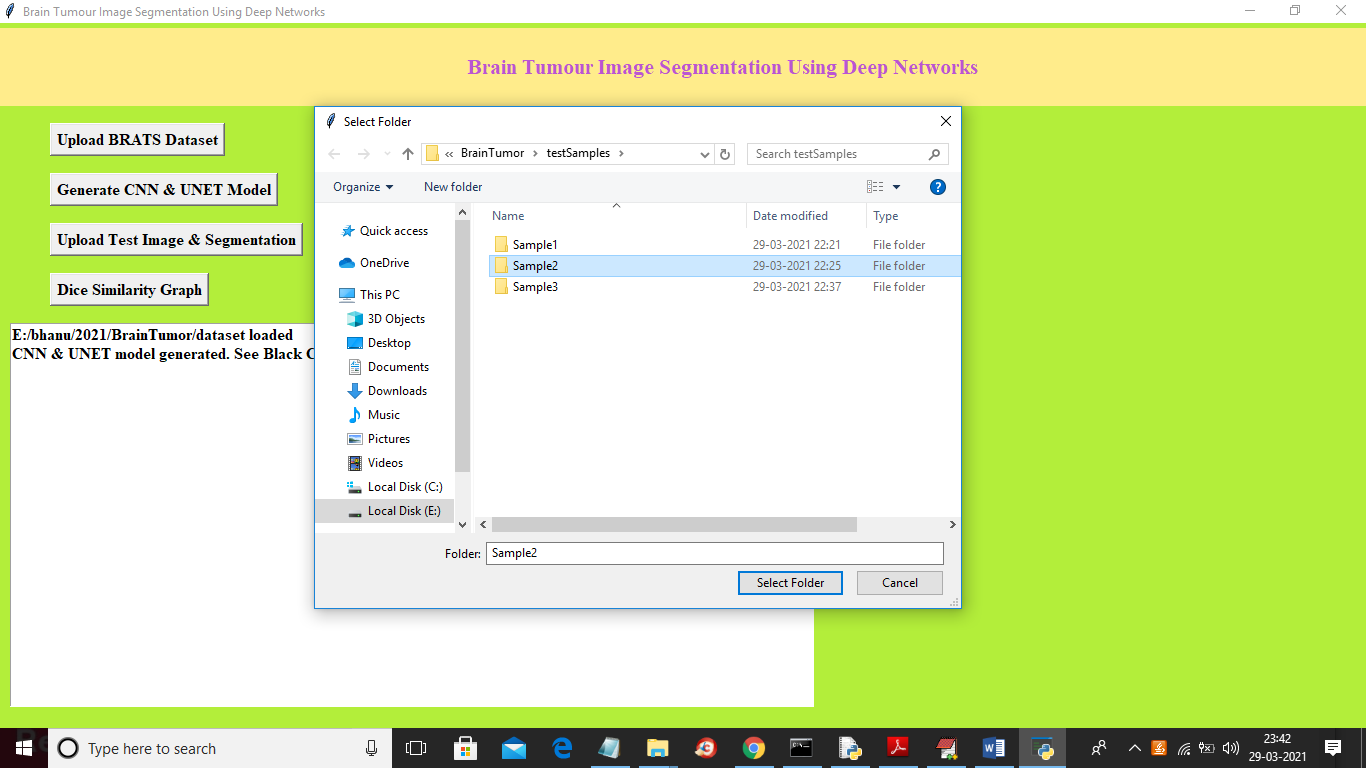
In above screen we can see models are using different size images to filter them and to get best features from it to build efficient model and now model is generate and now click on ‘Upload Test Image & Segmentation’ button and then upload test samples to get segmented output



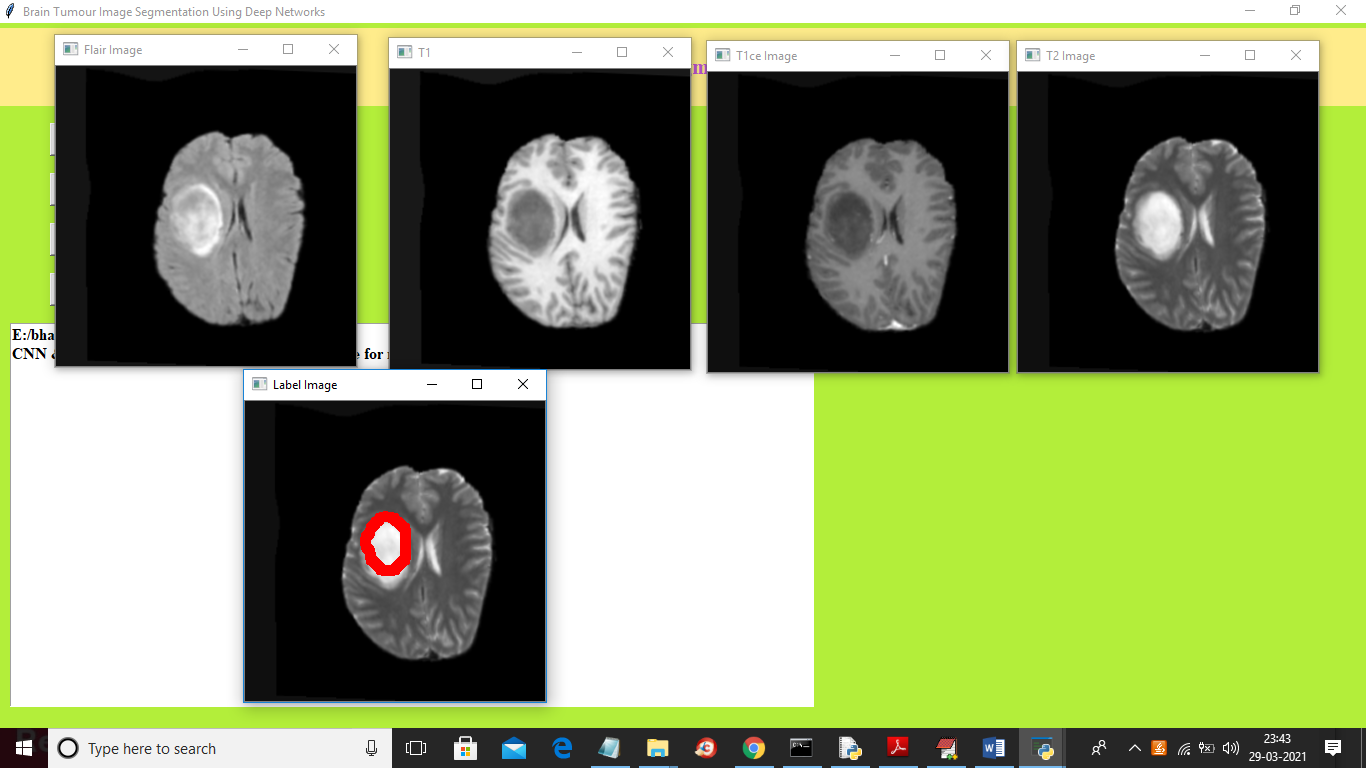
In above screen selecting and uploading ‘Sample1’ folder and then click on ‘Select Folder’ button to get below output



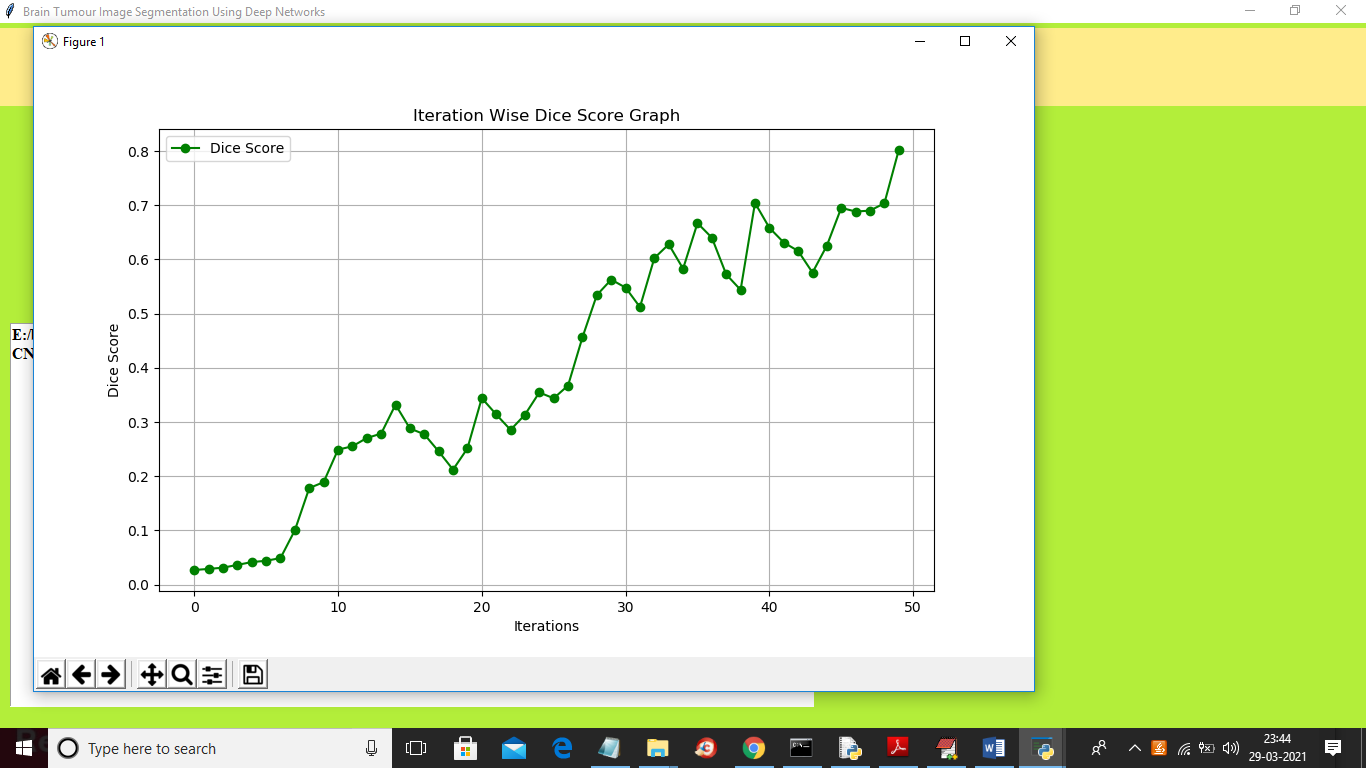
In above screen top 4 images are the input images such as FLAIR, T1, T2 and T1CE and 5th image is the predicted image with segmented part showing in red colour and this algorithm correctly detecting and marking tumour area and now test with other image



In above screen I am selecting and uploading ‘Sample2’ folder and then click on ‘Select Folder’ button to load images and to get below output



In above screen first 4 images are the input images and fifth image is the predicted label image with segmented parts around tumour area. Now click on ‘Dice Similarity Graph’ button to get below graph



To build CNN and UNET model we took 50 epoch or iterations and at each iteration DICE score between training and testing images get better and better and we get final dice score as 0.8 \* 100 = 80%. In above graph x-axis represents epoch and y-axis represents dice score