Segmentation and classification of brain tumor using 3D-UNet deep neural networks

Brain tumor is a deadly disease which causes death to millions every year and timely detection of such tumor can help in reducing risk of losing life. In the past many deep learning algorithms were introduced which can detect tumor and perform classification but its detection rate is low and work only 2 dimension MRI images. Latest technology generating MRI in 3D format and existing UNET segmentation cannot work on 3D MRI images and to solve this issue author of this paper employing 3D-UNET algorithm which will segment out tumor part from brain MRI and then employing 16 layer CNN algorithm to classify or damage brain tumor.

3D-UNET algorithm trained on BRATS2020 dataset to segment out tumor data and then propose 16 layer CNN algorithm trained on ‘Brain Tumor MRI Dataset’ which consists of 4 different classes listed below

'glioma', 'meningioma', 'notumor', 'pituitary'

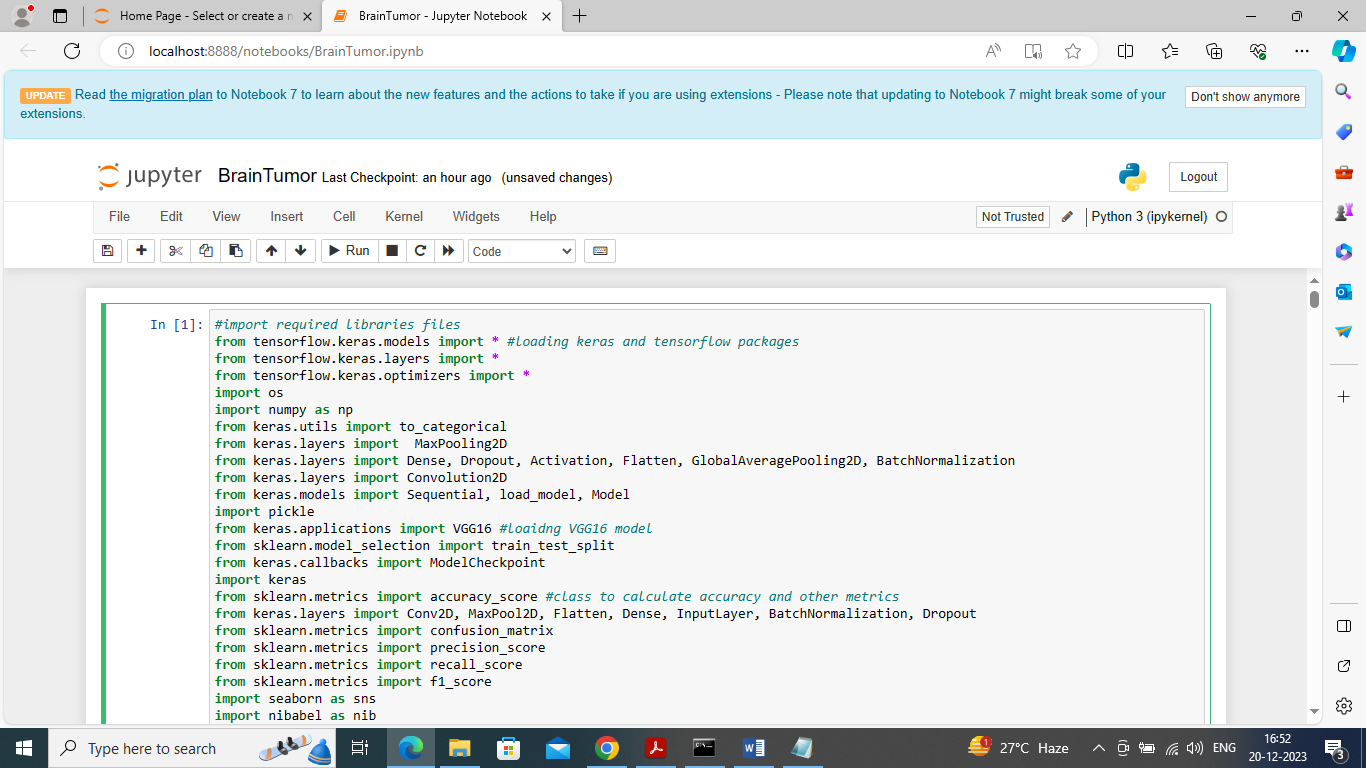
Above dataset can be download from below KAGGLE repository dataset

<https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>

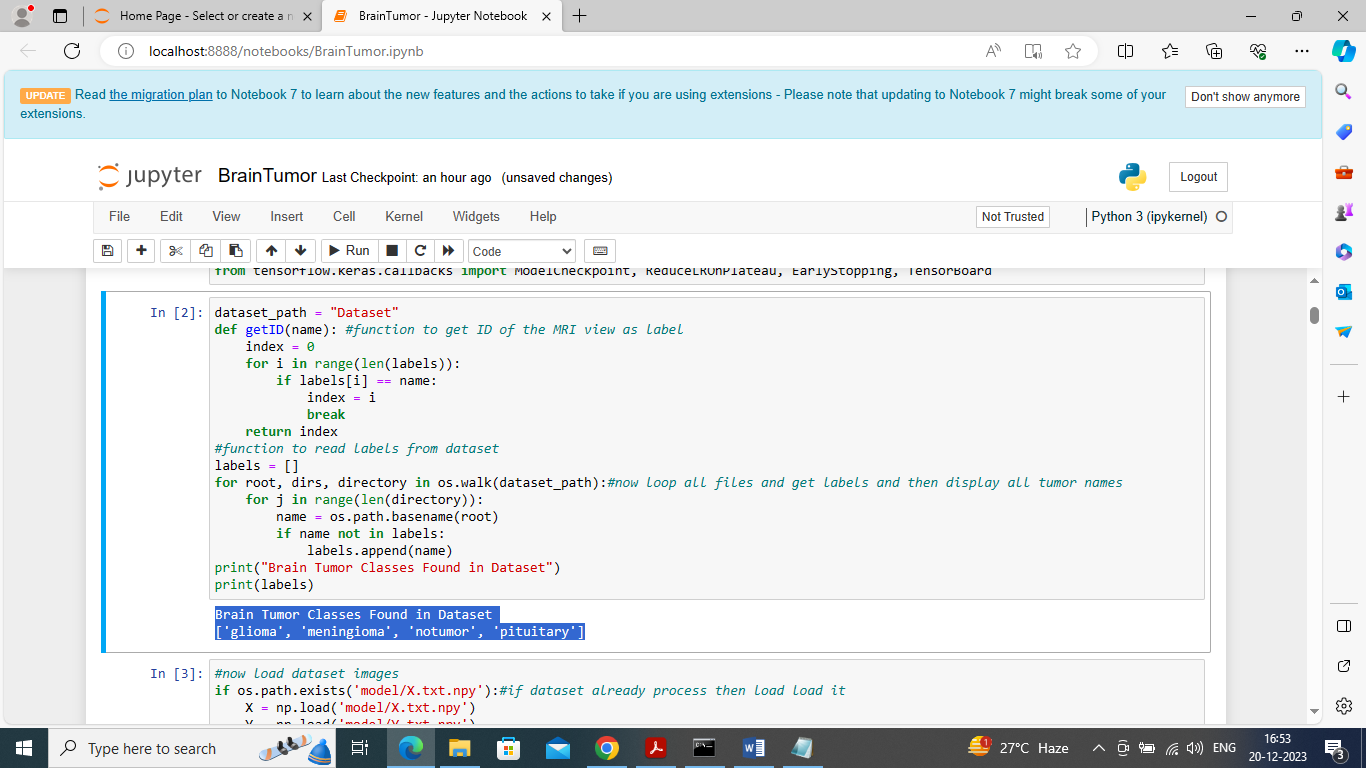
Above dataset trained on VGG16 pre-trained model and propose 16 layers CNN model and in both algorithm propose CNN 16 layer algorithm is giving best accuracy. Propose algorithm consist of CNN layer to filter MRI features and to efficiently extract tumor and then MaxPool2d layer will collect filtered features from CNN and then apply Dropout layer to remove irrelevant features. This filtration make propose CNN algorithm to detect and classify tumor 90% accurately.

3D-UNET algorithm can able to train and segment tumor part from 3D images and by seeing this segmented tumor output doctors can easily identify tumor region and based on region they can perform suitable treatment to reduce risk of patient life.

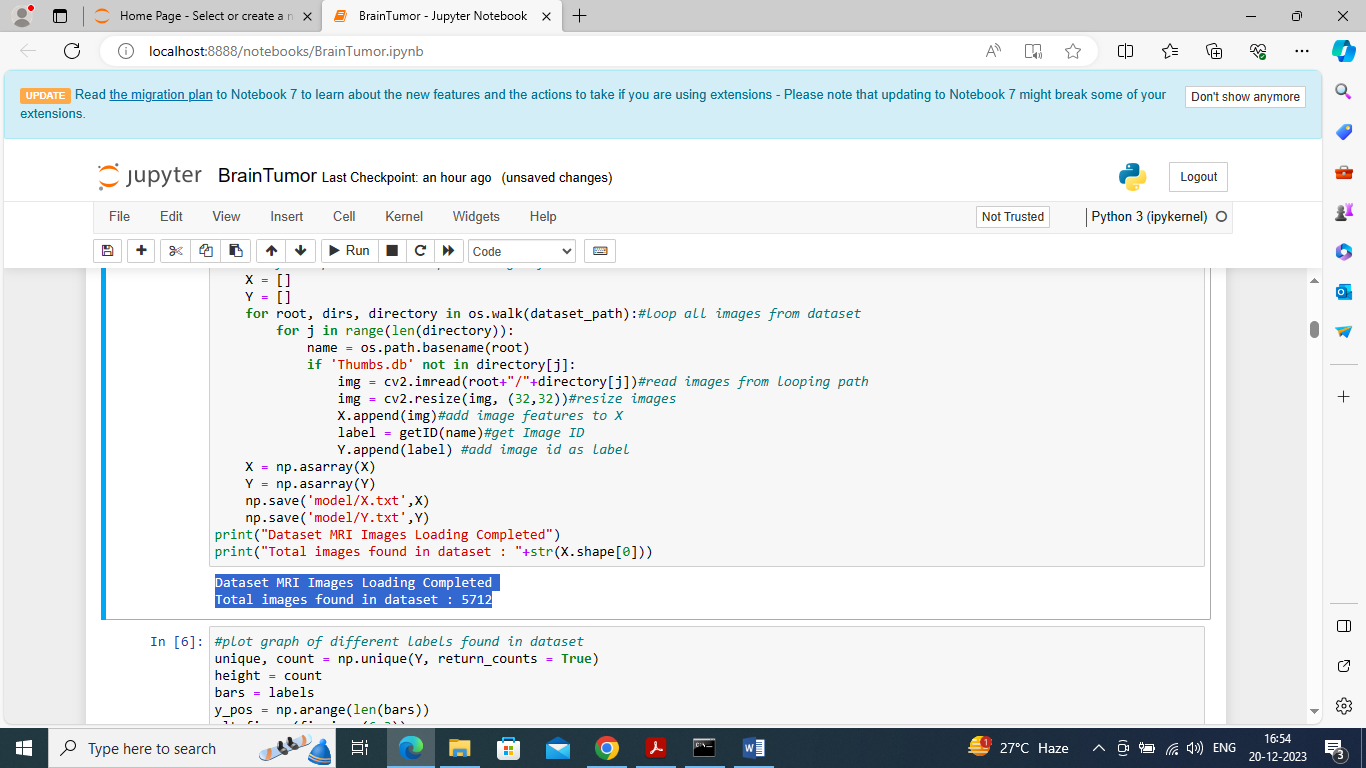
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



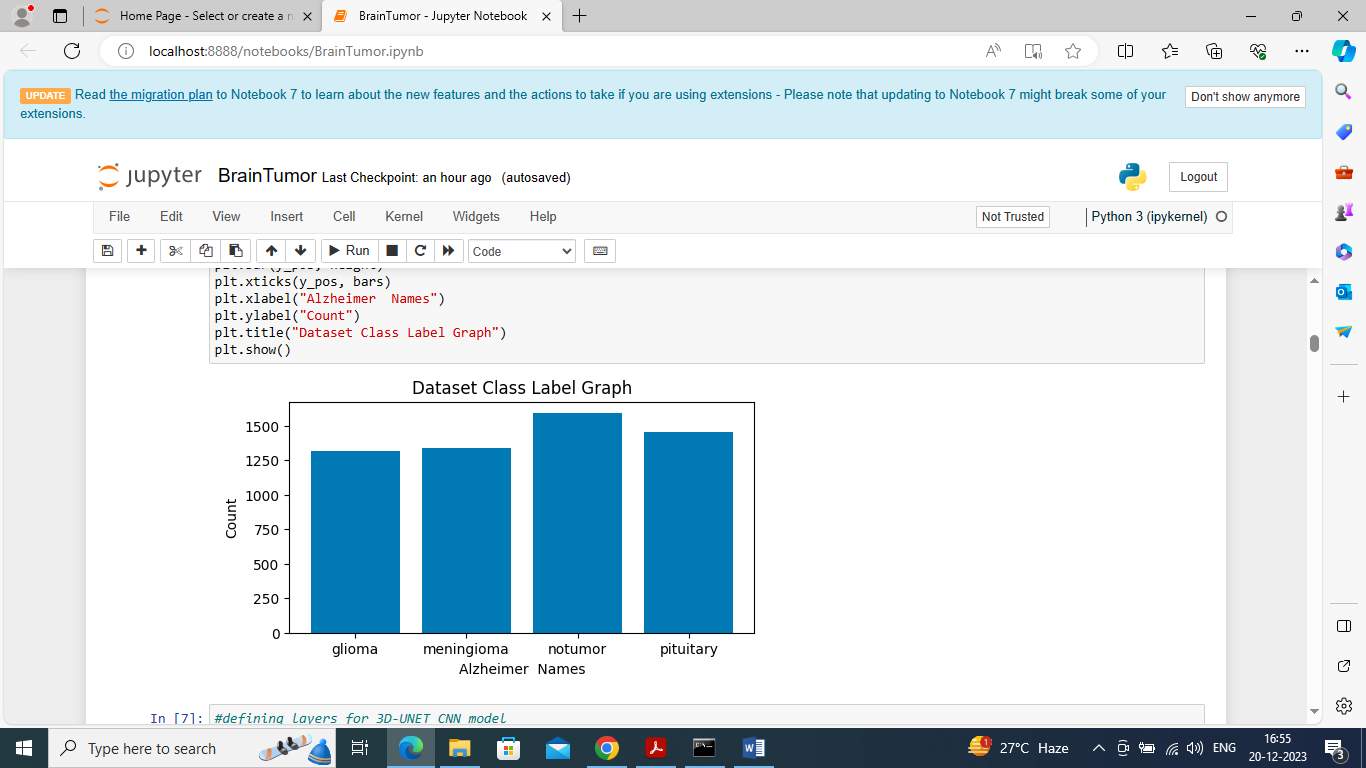
In above screen importing required classes and packages



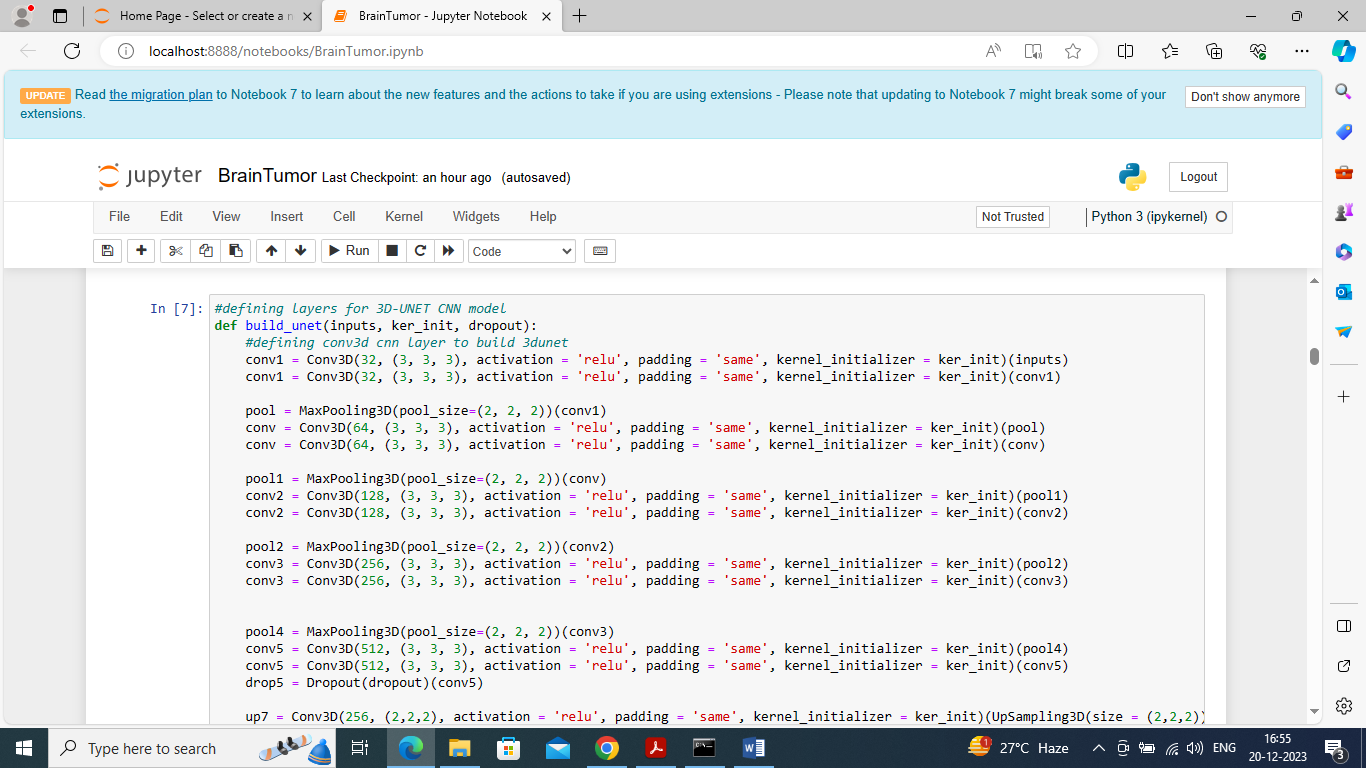
In above screen loading dataset to identify different tumor or class labels found in dataset and all tumor names we can see in blue colour text



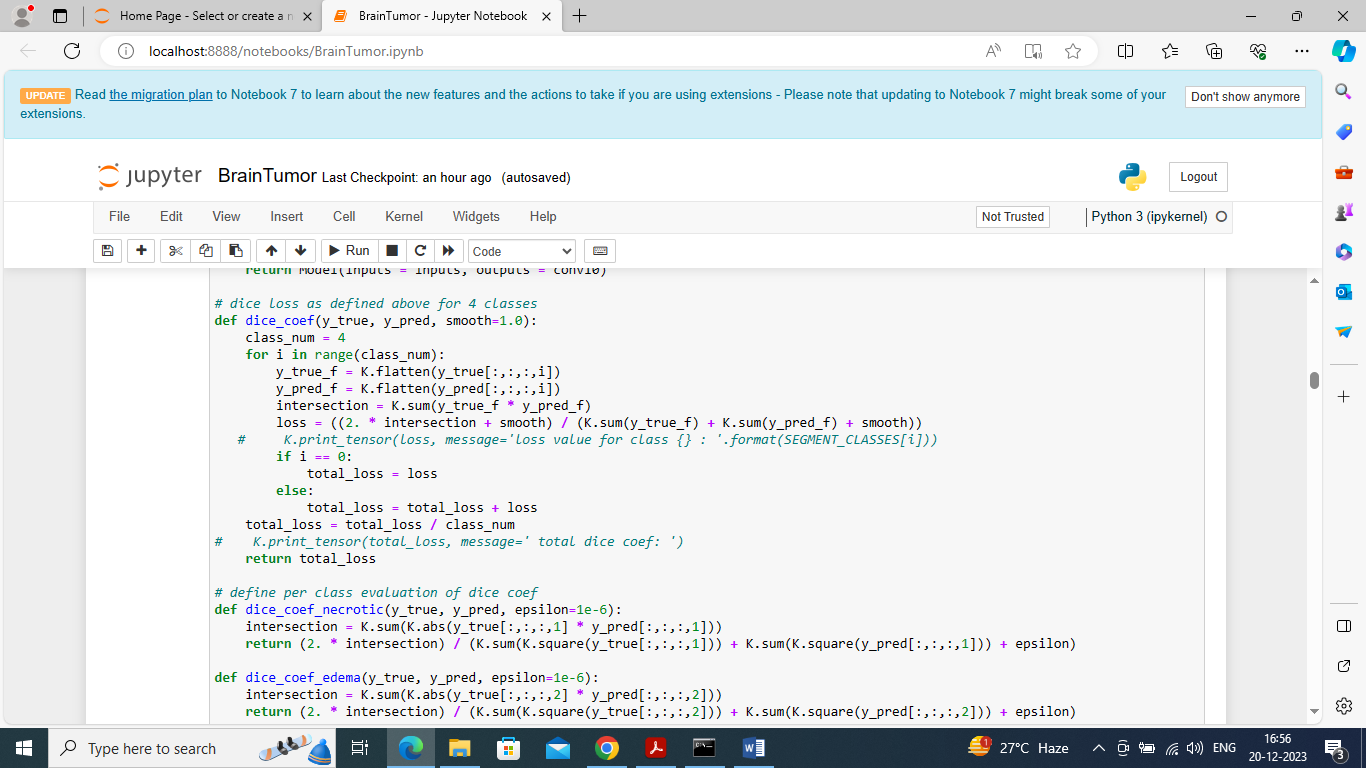
In above screen looping and reading all images from dataset and then resizing all images to equal size and then adding to X and Y training arrays and then in blue colour text can see total number of images loaded



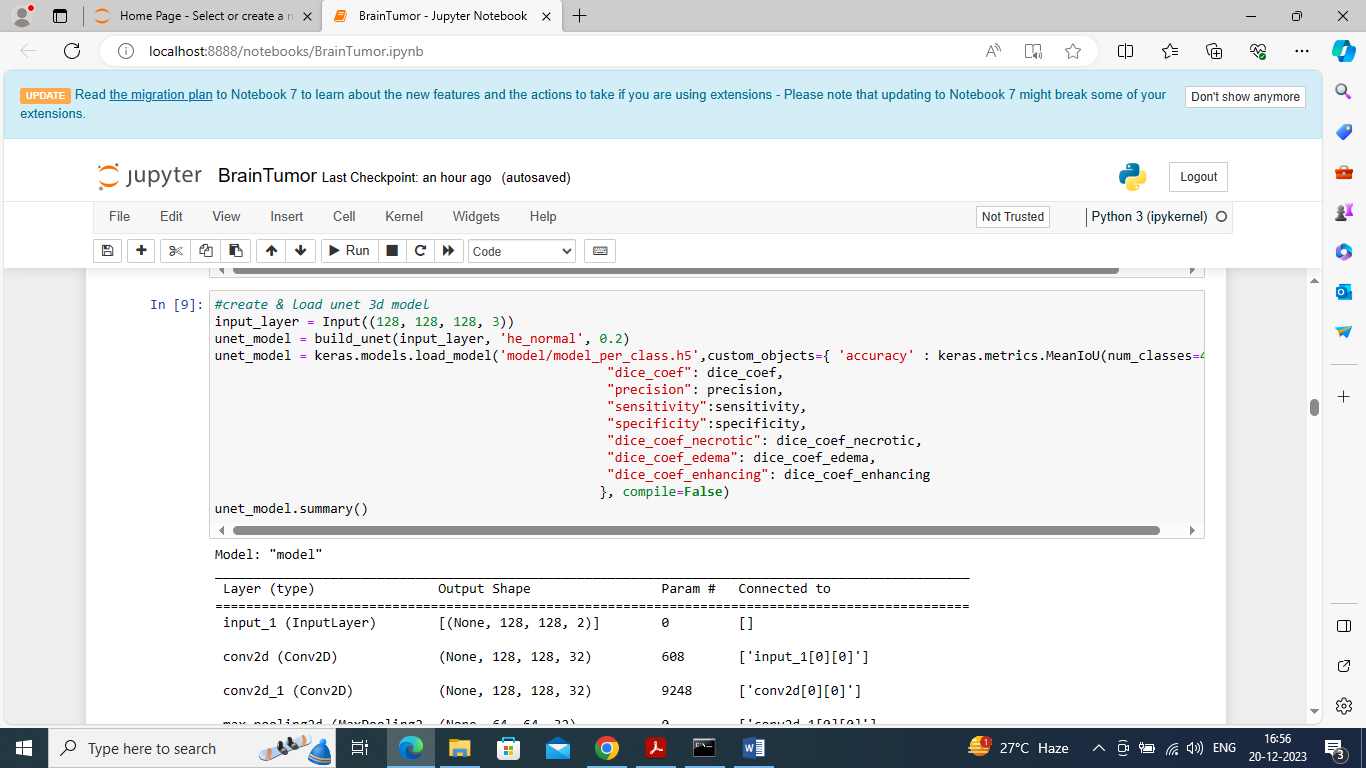
In above graph x-axis represents tumor class label and y-axis represents number of images found in that class label



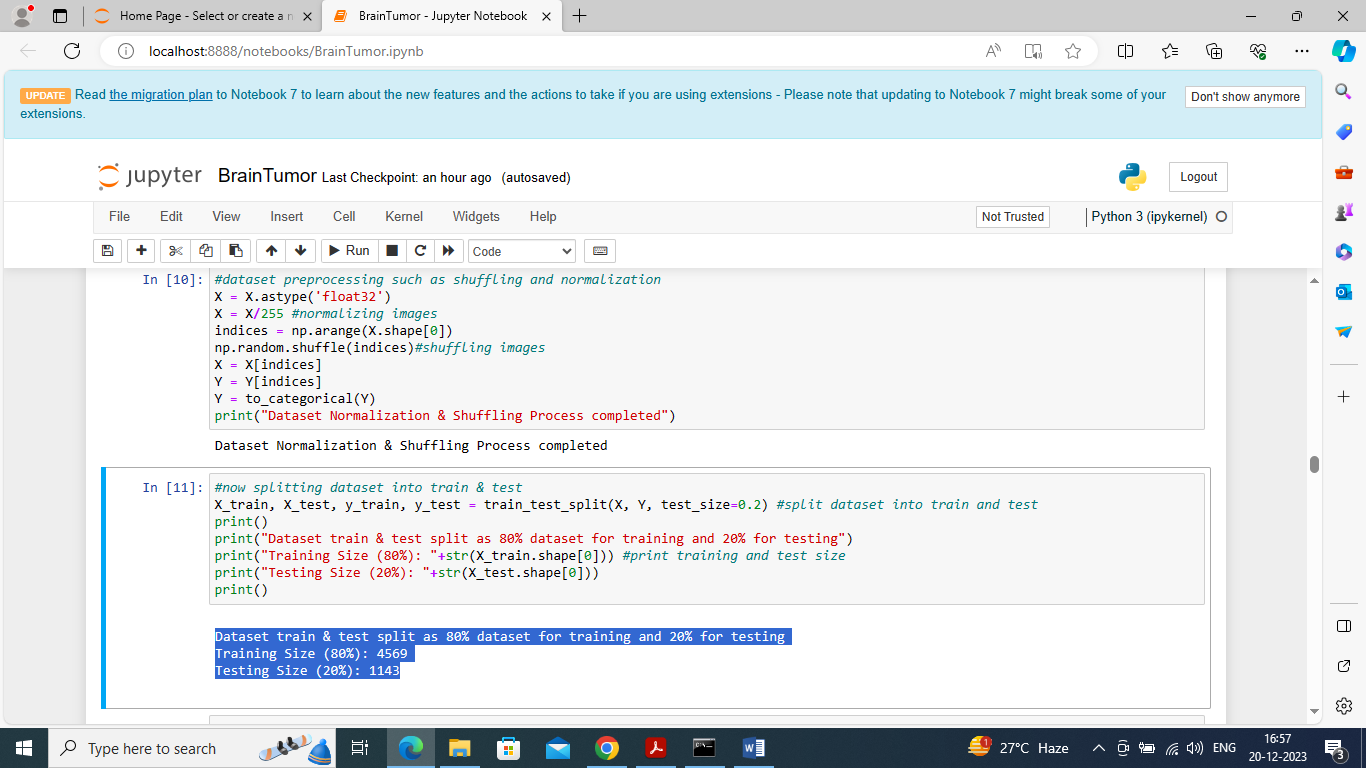
In above screen defining 3D-UNET model by using CONV3D layer



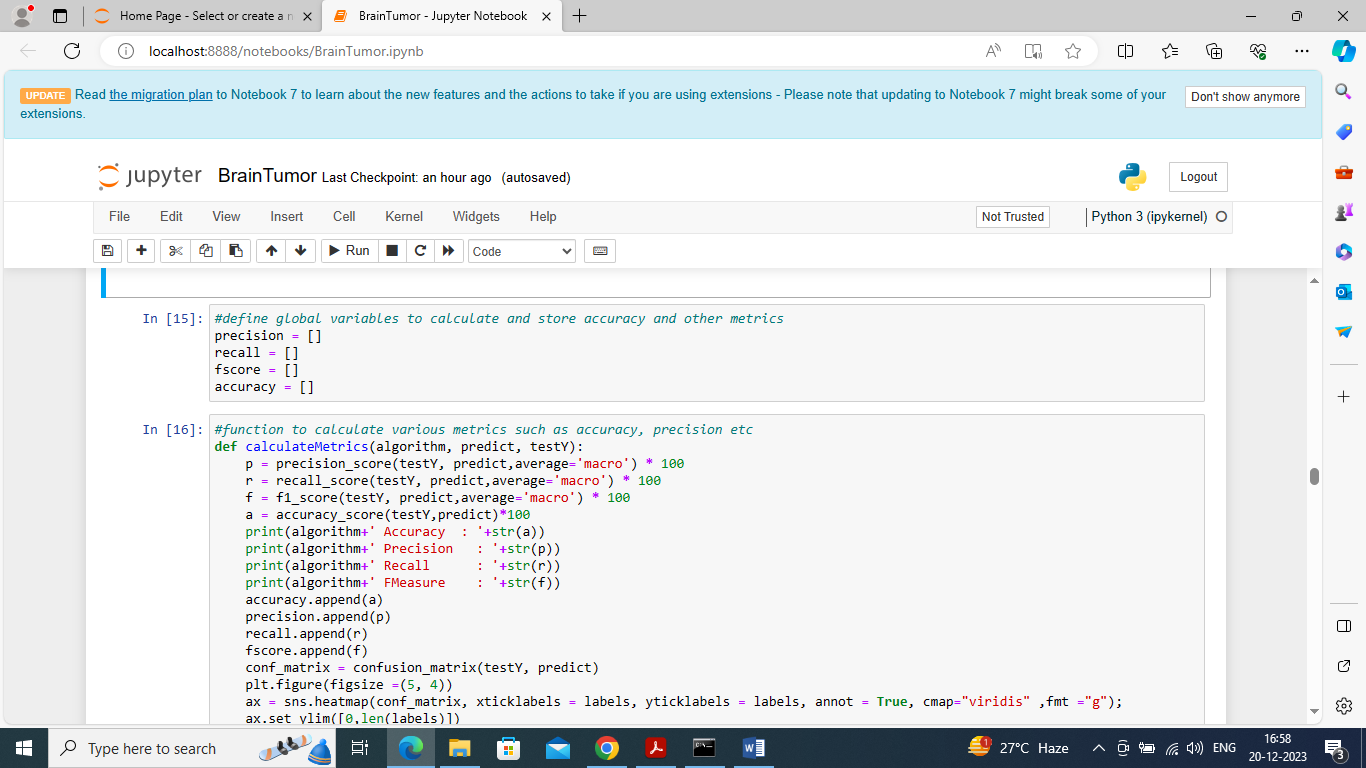
In above screen defining dice score function to train UNET with dice score



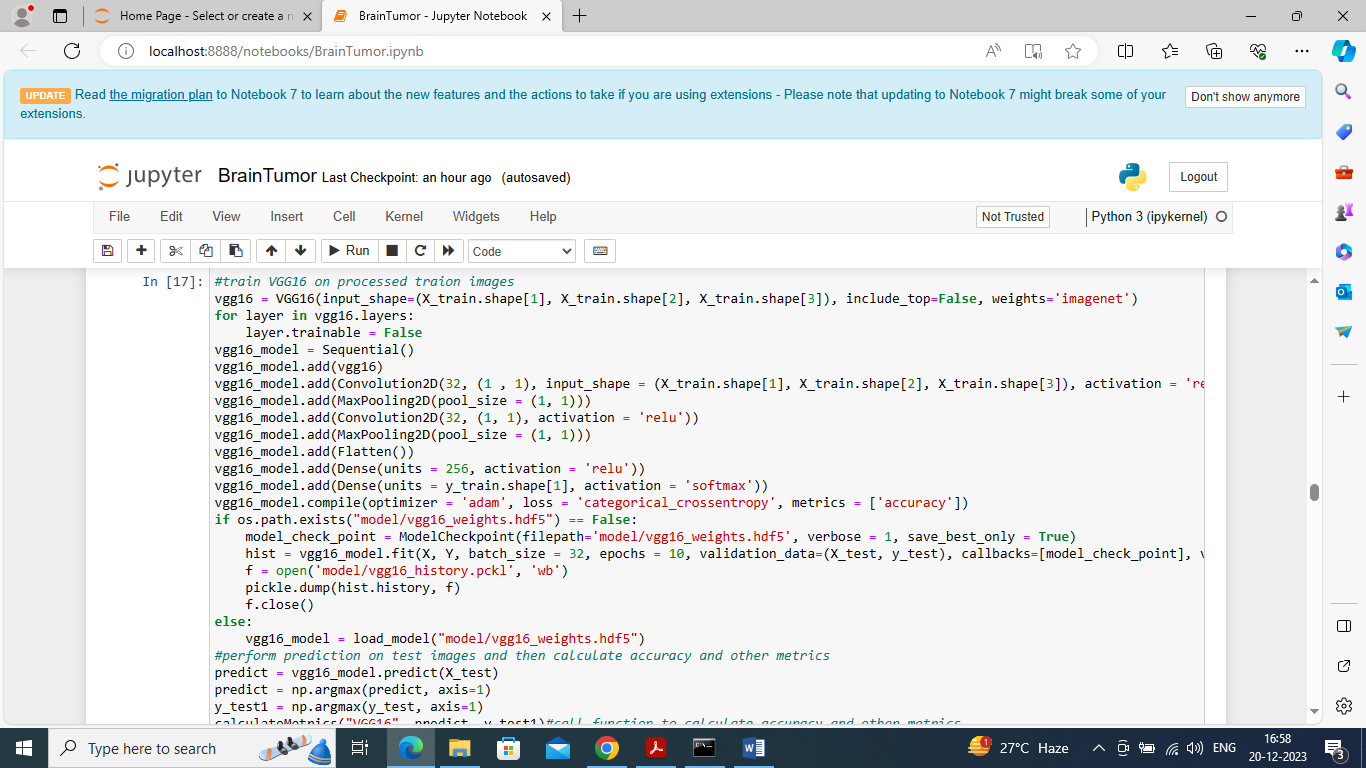
In above screen creating and loading 3DUNET model with all metrics like Precision, dice and many more and then displaying loaded model details



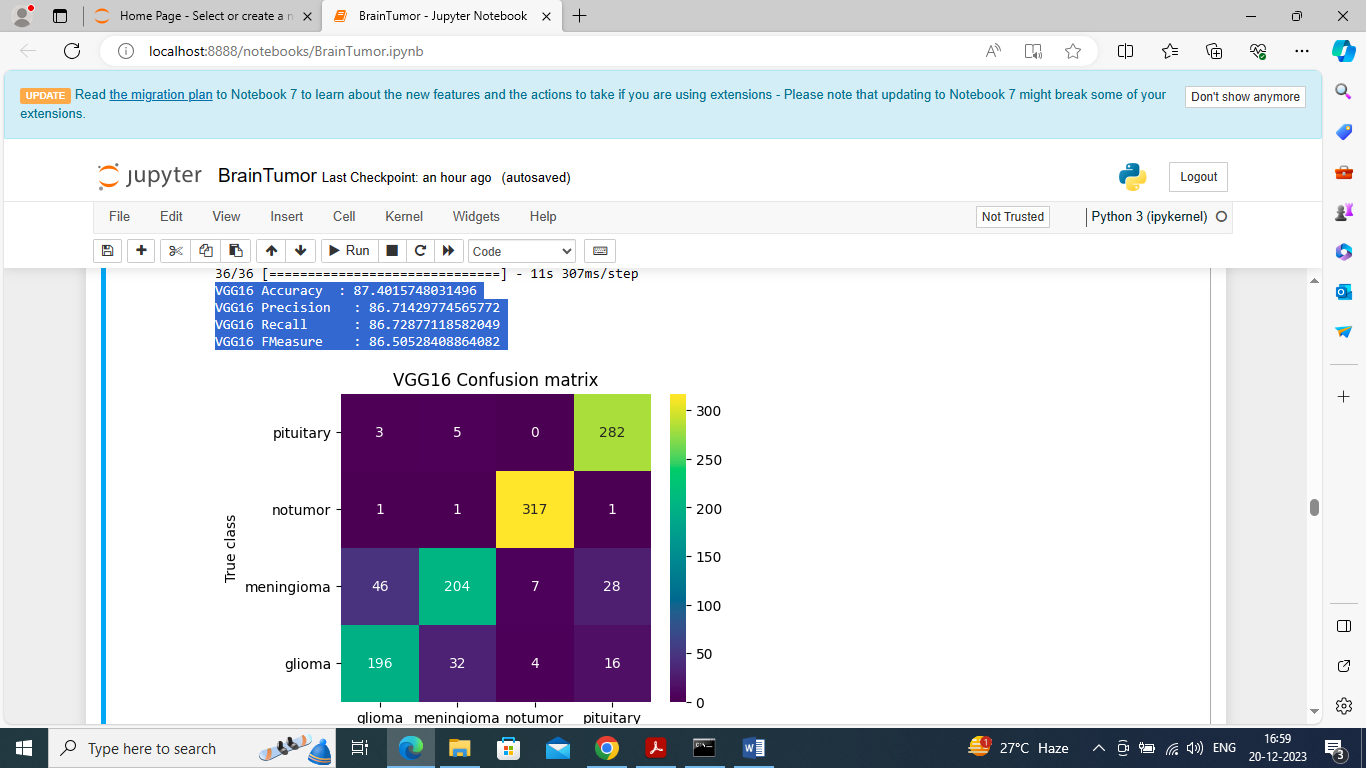
In above screen defining code to shuffle, normalize images and then split images into train and test where application using 80% images for training and 20% for testing



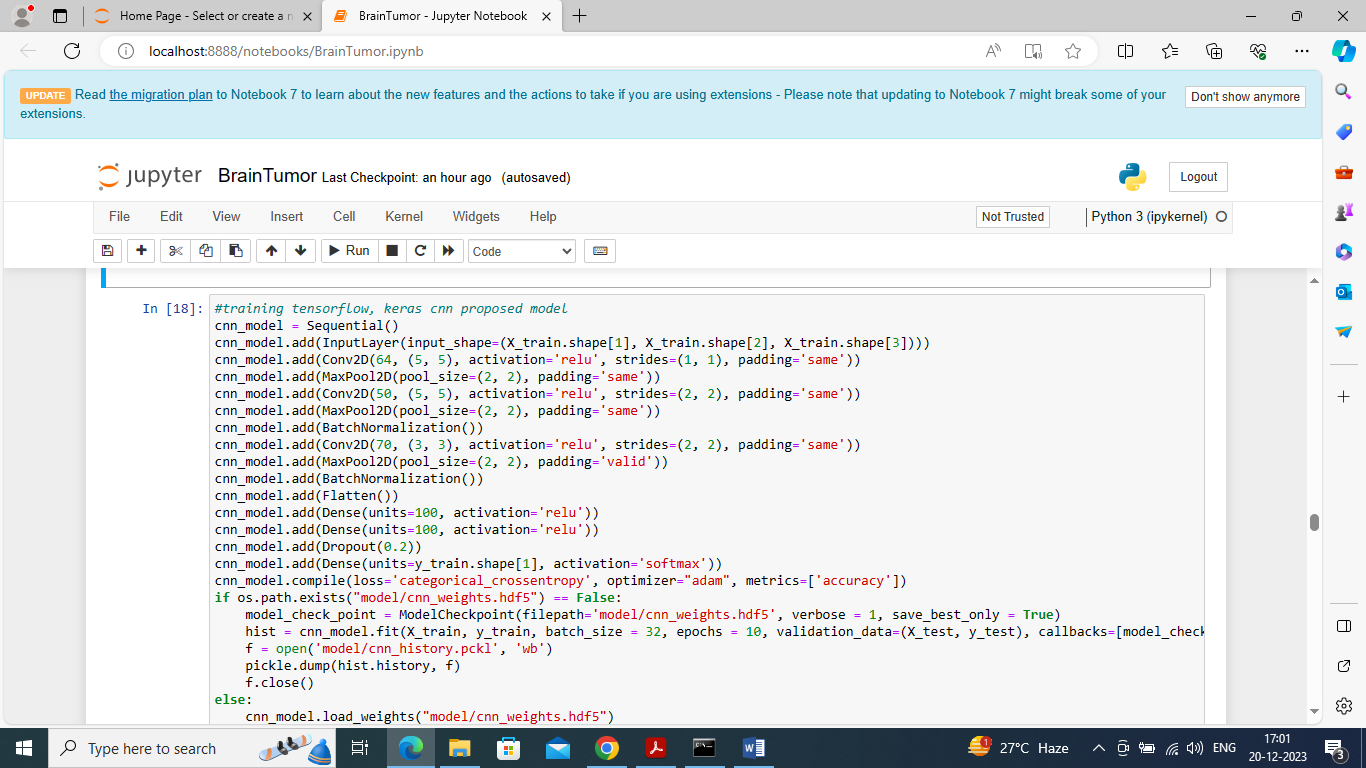
In above screen defining function to calculate accuracy and other metrics



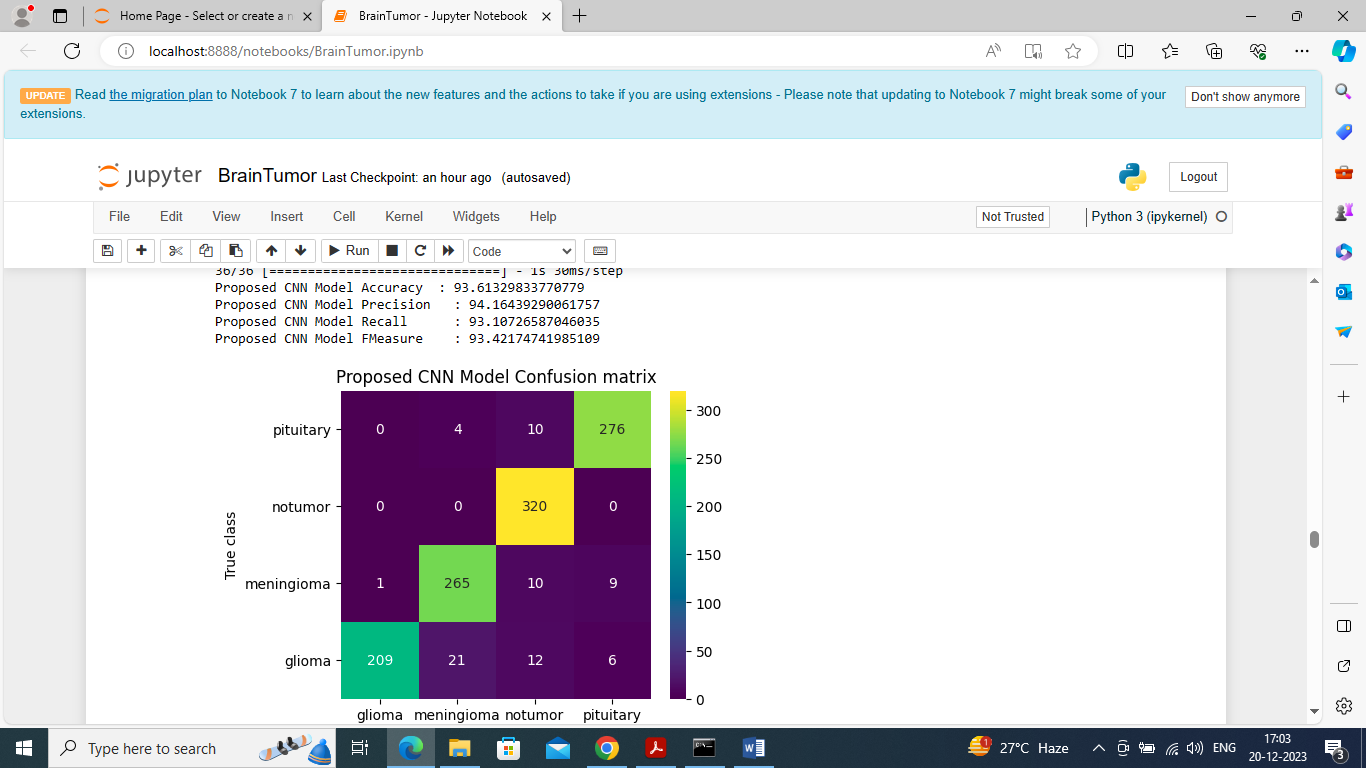
In above screen training VGG16 existing algorithm and after executing above model on 20% test images will get below output



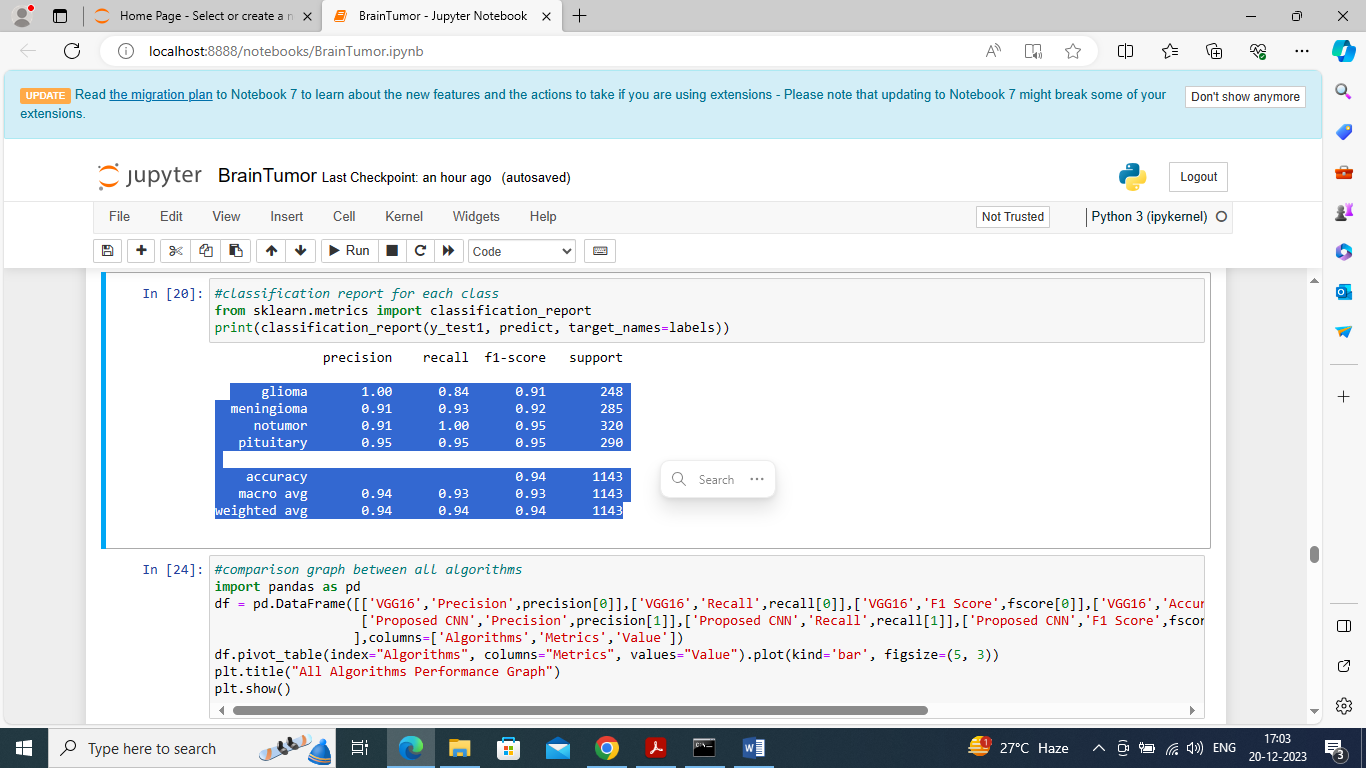
In above screen VGG16 got 87% accuracy and can see other metrics like precision, recall etc. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents true labels and all boxes count if diagnol with different colour represents Correct Prediction count and remaining blue boxes represents incorrect prediction count which are very few



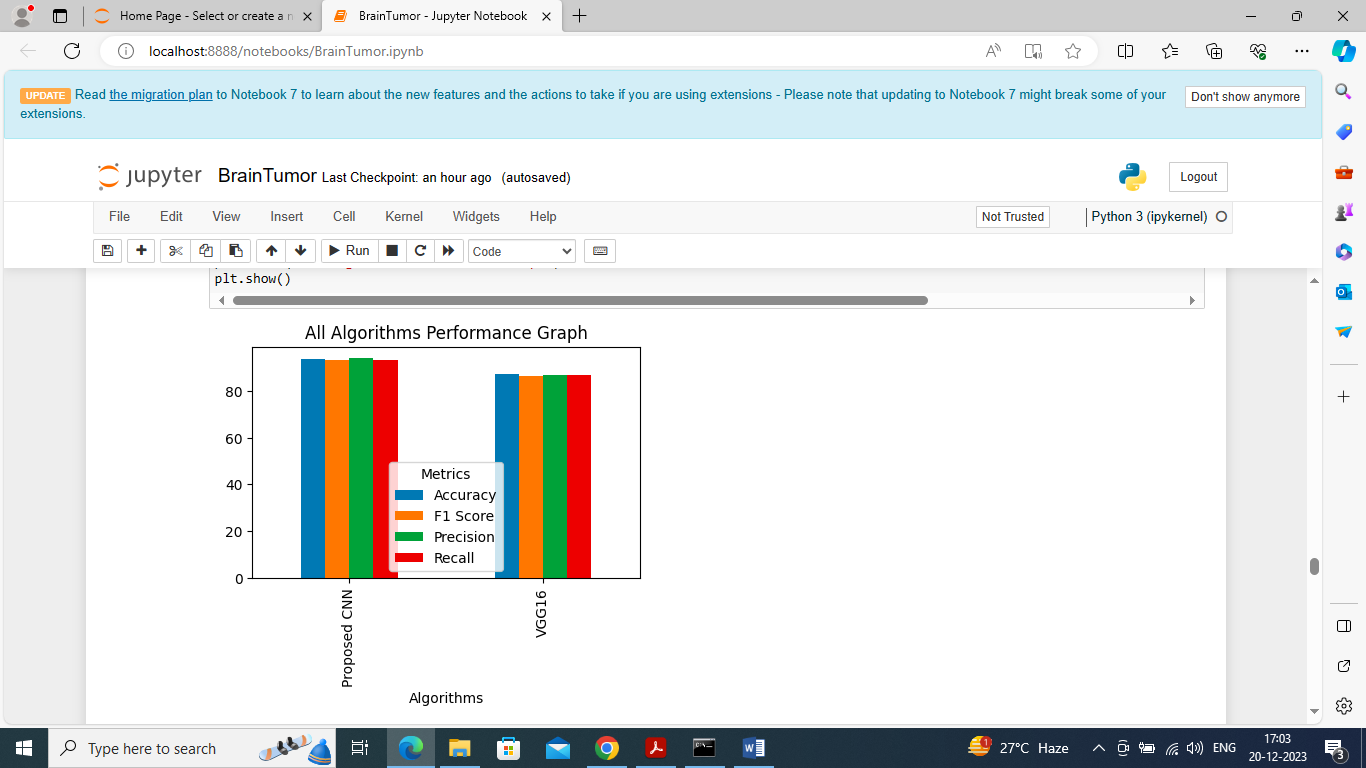
In above screen defining propose 16 layers CNN model and in above code each line will represents one layer and after executing above model will get below output



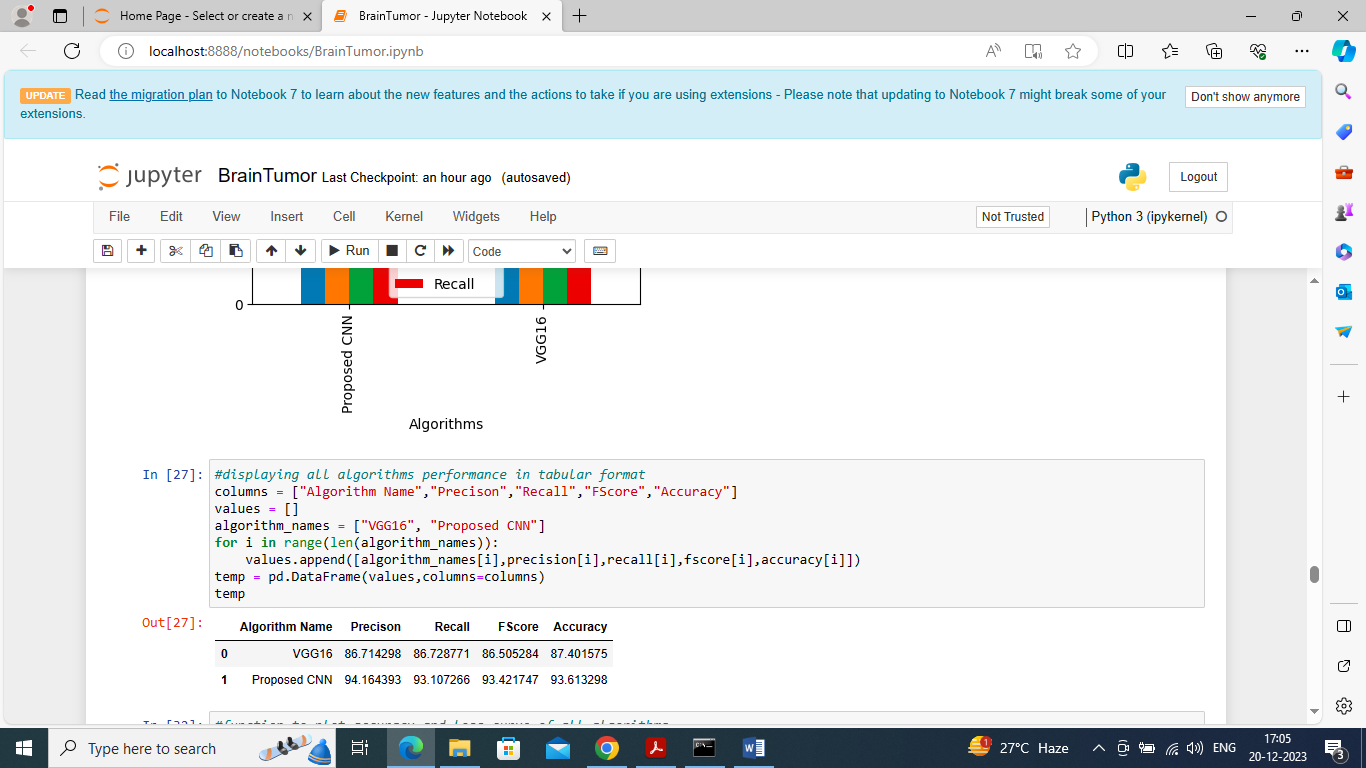
In above screen propose CNN model got 93% accuracy and can see other results also



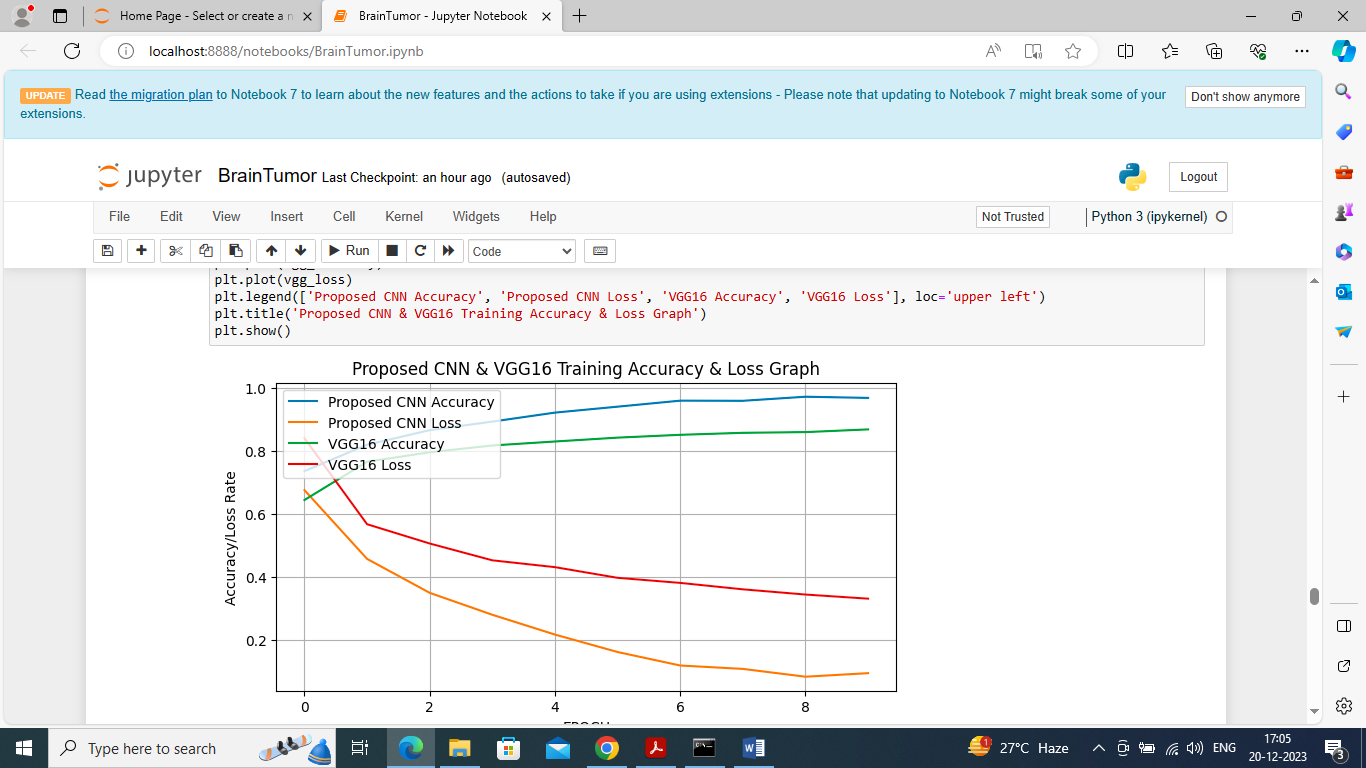
In above screen can see classification report output for each tumor class



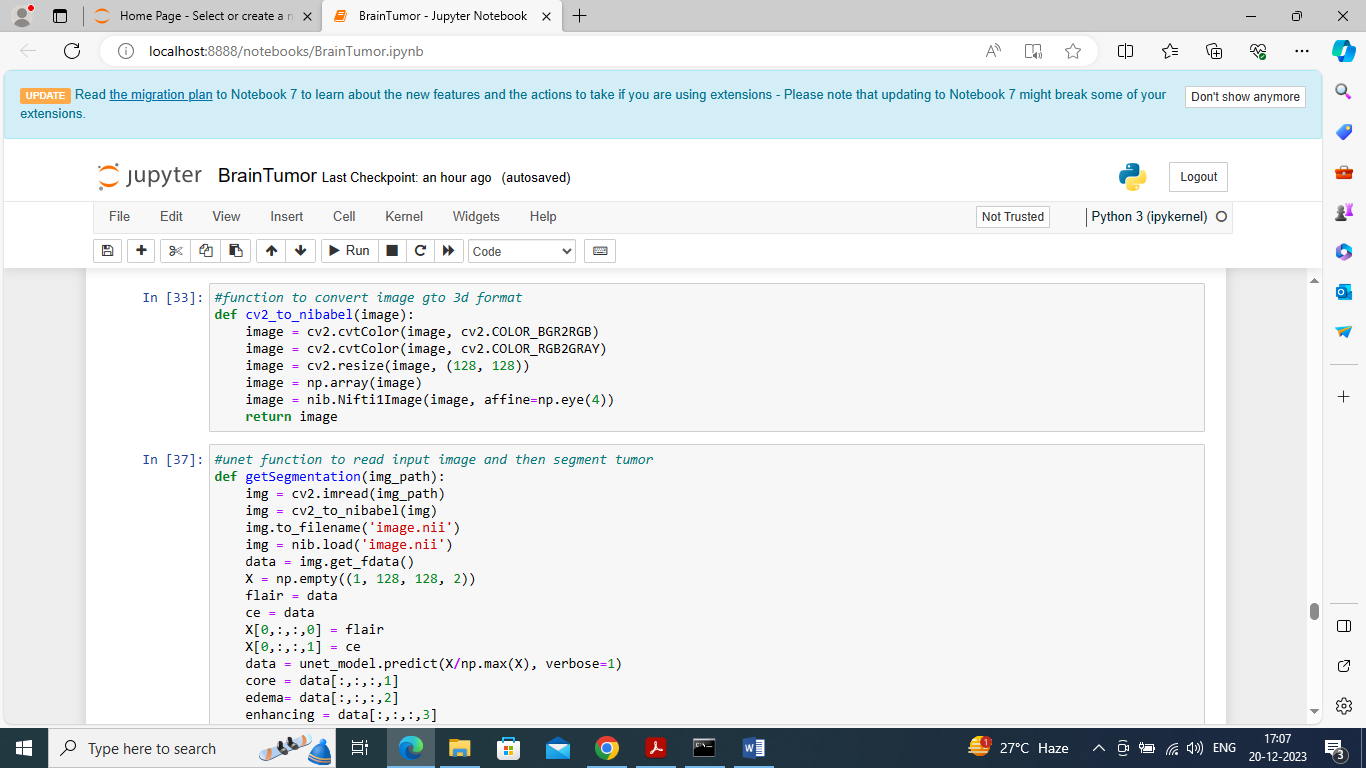
In above screen displaying comparison graph between propose and VGG16 and in above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in above graph in both algorithms propose got high results



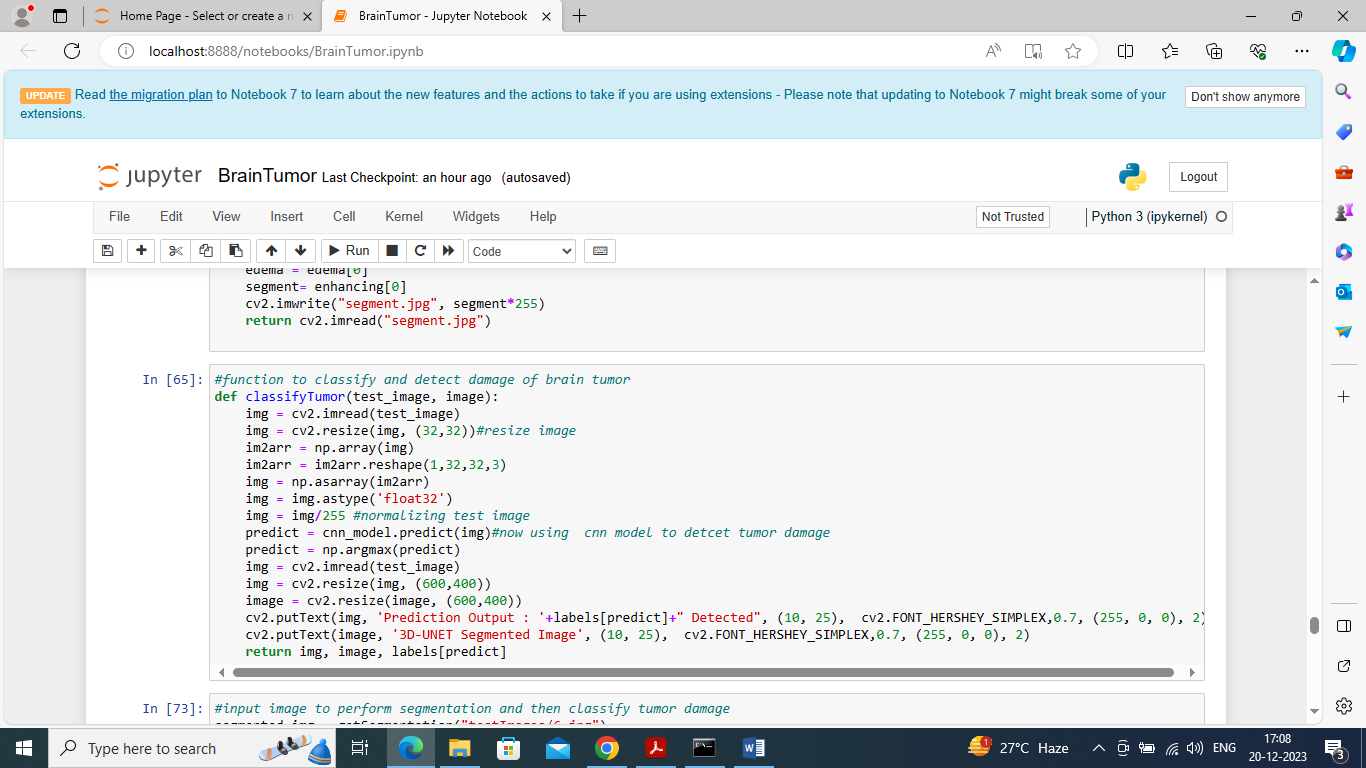
In above screen can see both algorithm results in tabular format



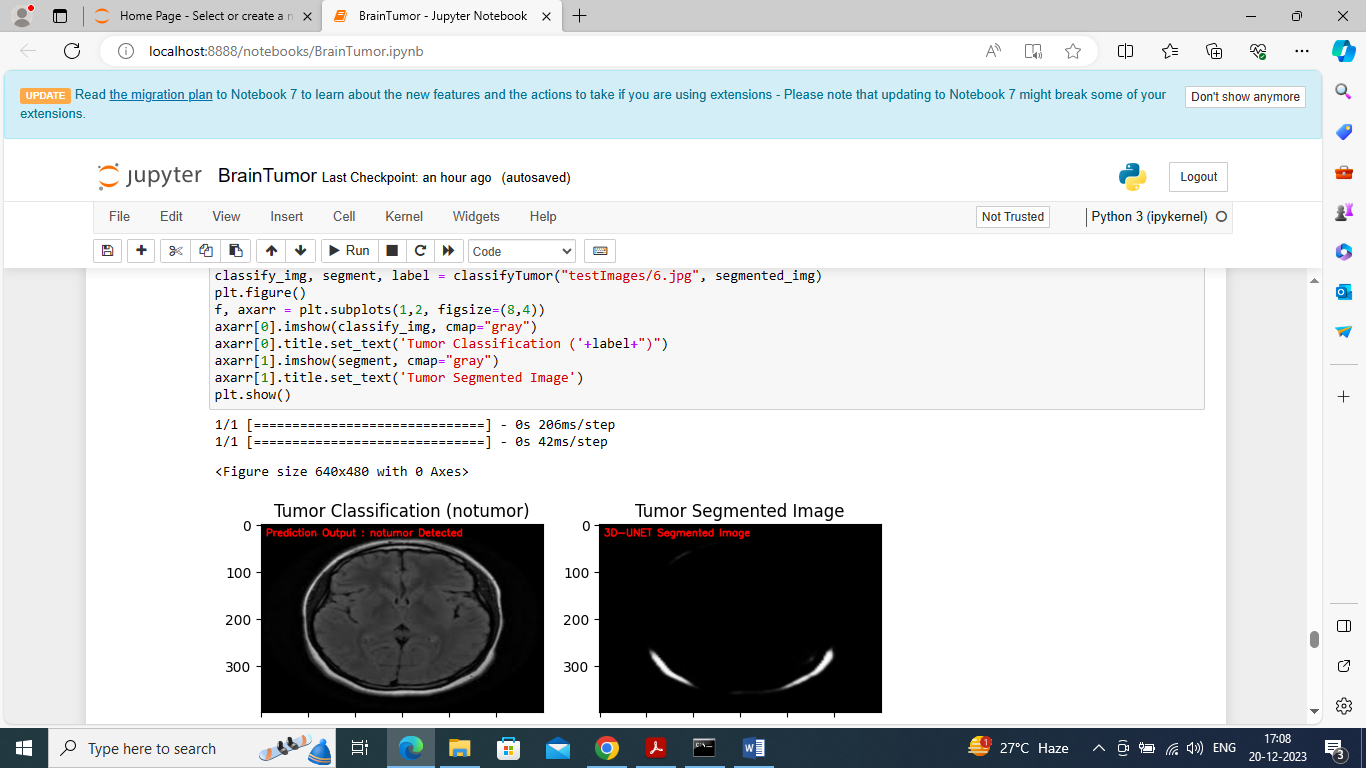
In above graph displaying VGG and propose CNN training and loss graph and in graph x-axis represents training epochs and y-axis represents accuracy/loss values and then blue line is for propose CNN accuracy and green line for VGG16 accuracy and red line for VGG16 loss and orange line for propose CNN loss. In above graph we can see with each increasing epoch accuracy got increase and loss got decrease for both algorithms but propose got high training accuracy and less loss.



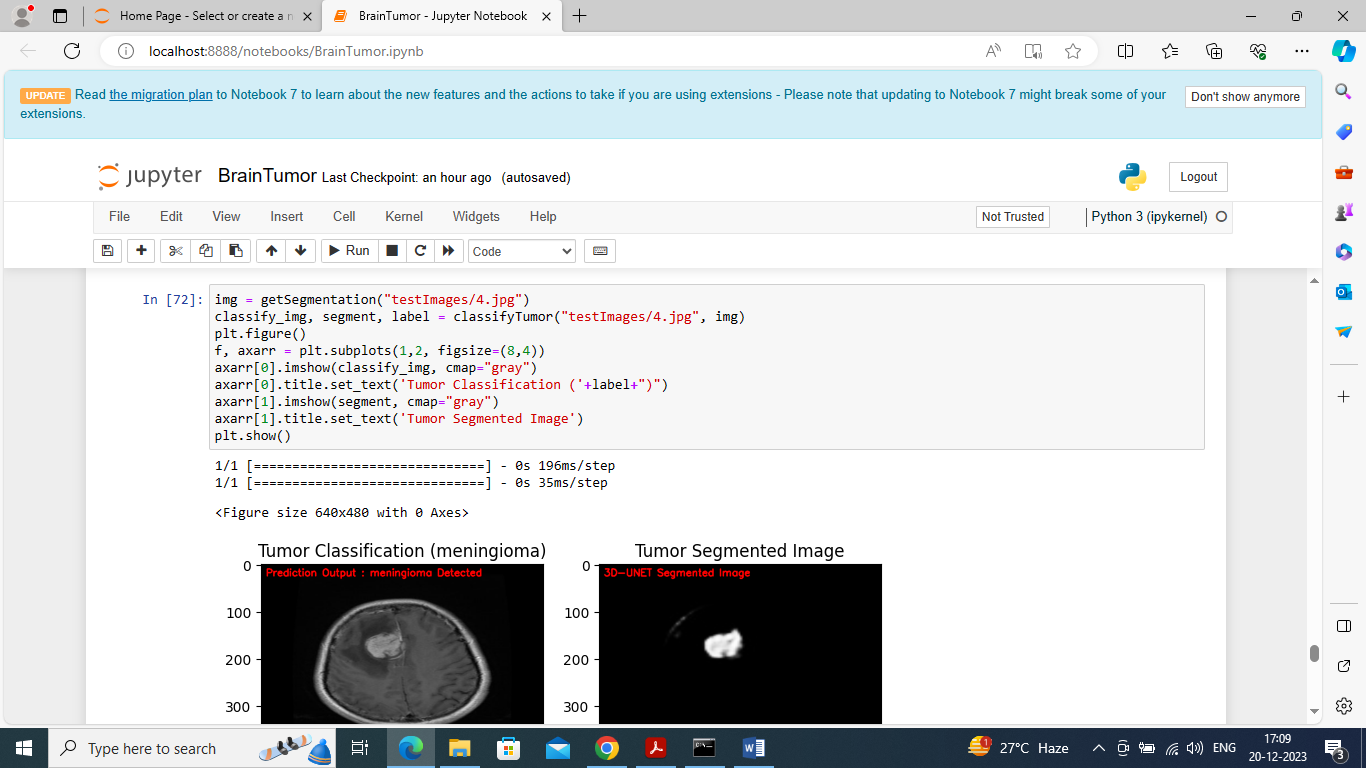
In above screen defining function to segment test image using UNET



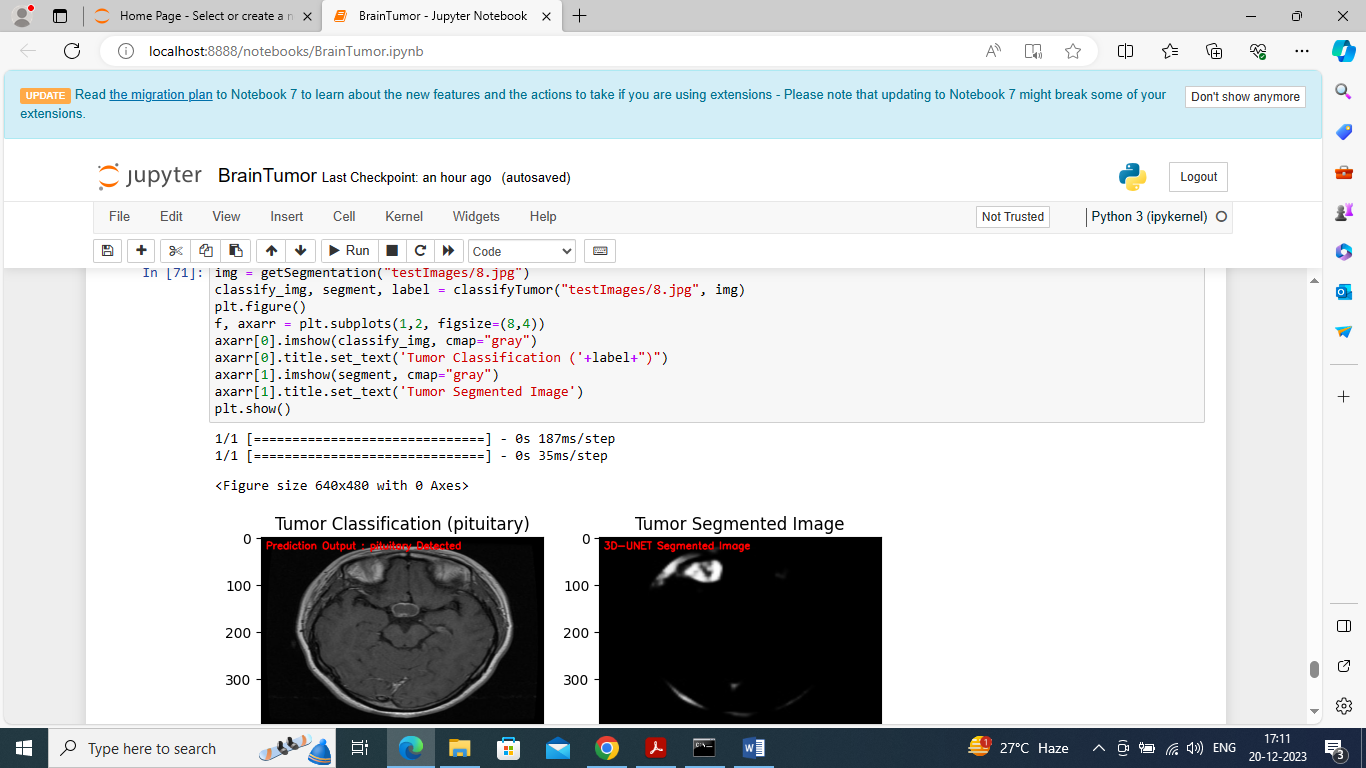
In above screen defining function to classify tumor or predict damage



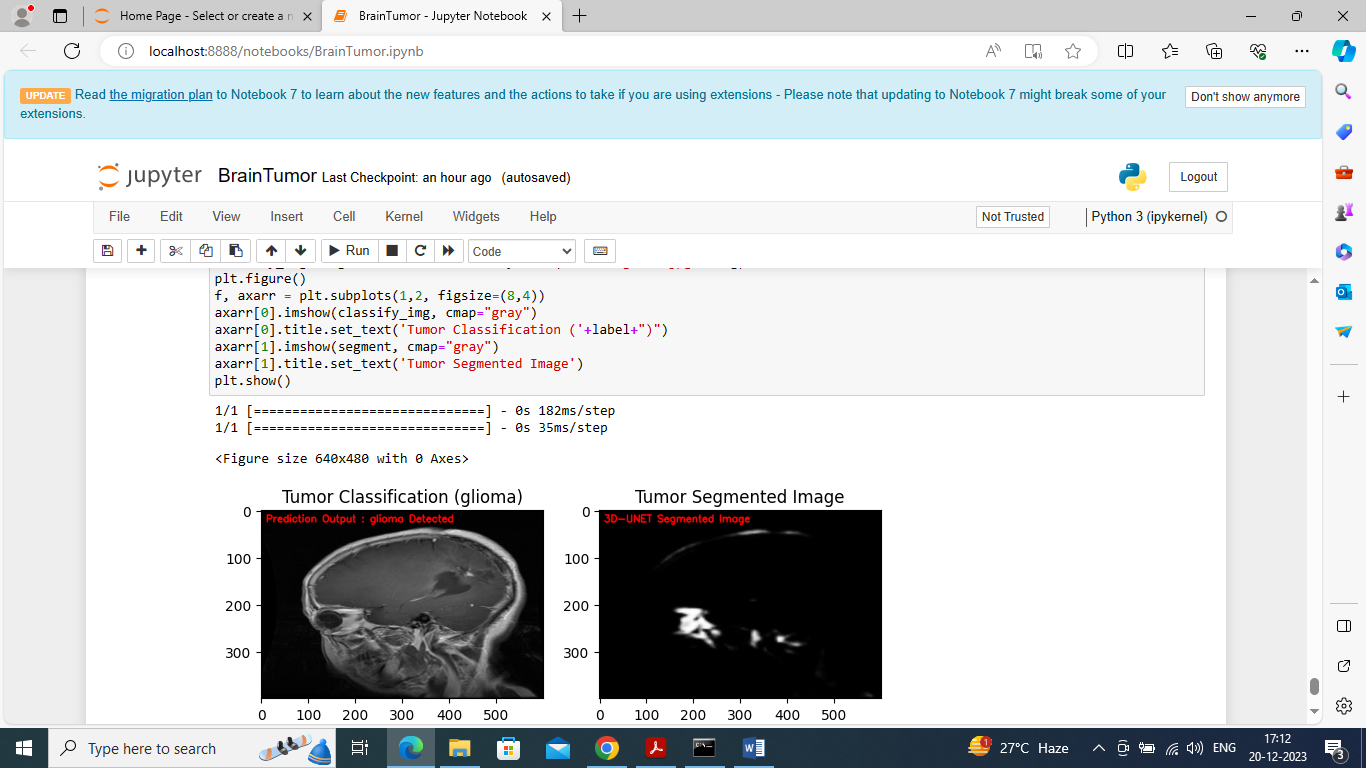
In above screen giving test image for segmentation and in output given image predicted as ‘NO Tumor’ and in segmented image also we cannot see any tumor region so brain is normal. Predicted output you can see in red colour text or in image title



In above screen ‘meningioma’ tumor is predicted and in segmented output we can see that tumor clearly. Segmented image showing in second part of image



In above screen ‘pituitary’ tumor detected and in segmented second image we can see tumor clearly



In above screen ‘glioma’ tumor detected and in segmented second image we can see tumor part