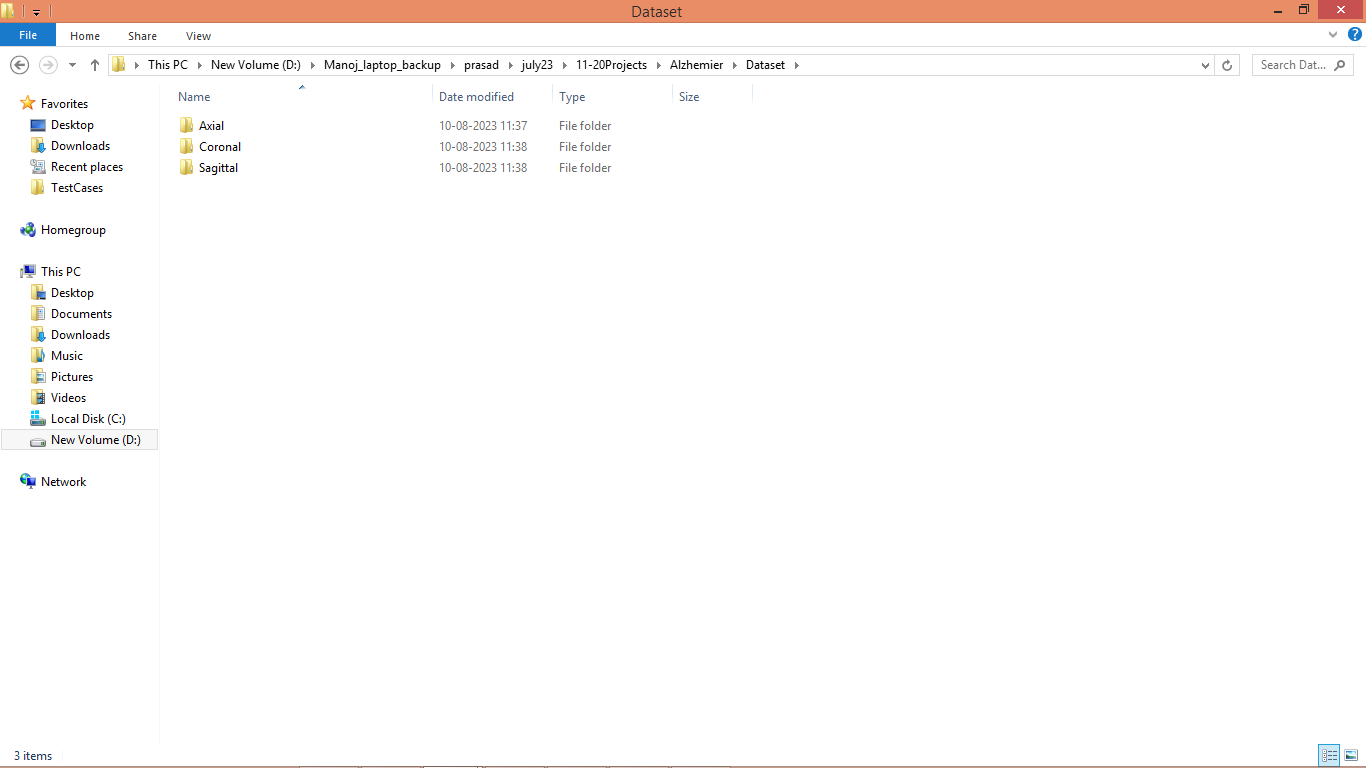
Diagnosis of Alzheimer’s Disease Using Convolutional Neural Network With Select Slices by Landmark on Hippocampus in MRI Images

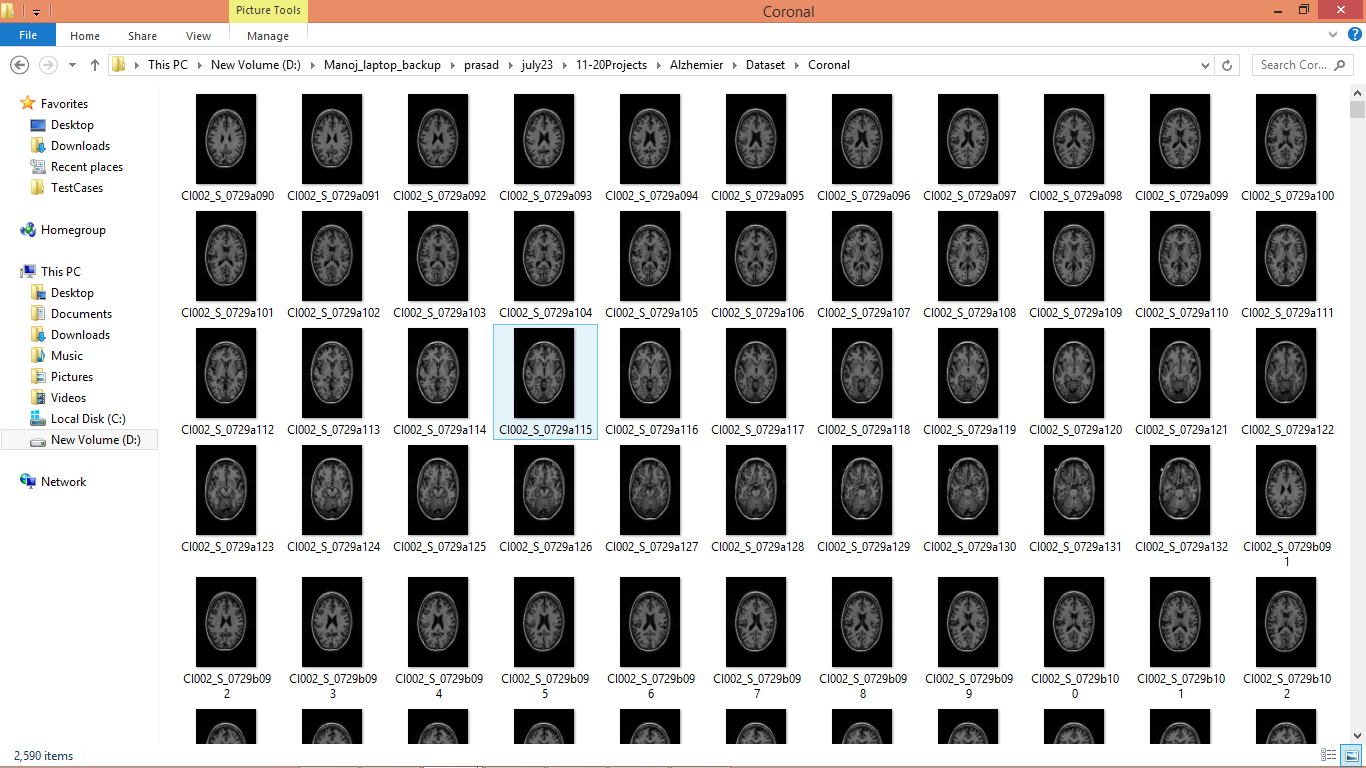
Alzheimer is brain related diseases which must be diagnose and treat in early stages to save patient life. In the past many machine and deep learning algorithms were using to predict this disease but all those algorithm were taking entire MRI scan input for prediction which is degrading prediction accuracy as entire image may contains irrelevant features which will be diagnose incorrectly. In Alzheimer patient Hippocampus is the main region which is mostly affected so author of this paper employing novel technique to extract affected regions such as Axial, Coronal and Sagittal from MRI scan images and this extracted brain images region will get trained with different deep learning algorithms such as Pre-trained Resnet50, Resnet50 and LENET and in all algorithms LENET is giving better accuracy.

In propose paper author has extracted regions manually by using experts from ADNI dataset and not publish on internet so we have downloaded available dataset images from GOOGLE and then train with above mention algorithms.

In below screen showing dataset details



In above screen we have 3 folders for each region view and just go inside any folder to view related images



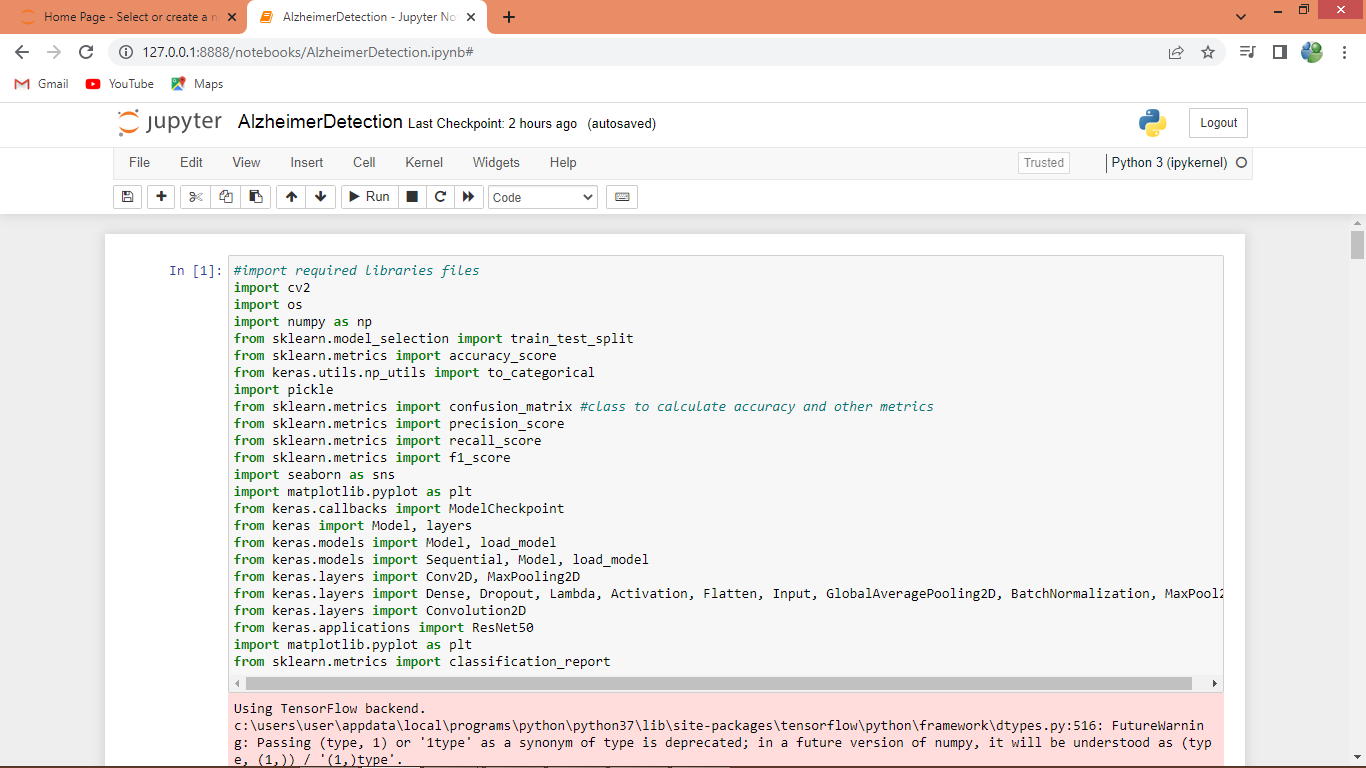
In above screen we can see images used to train models.

Extension Concept

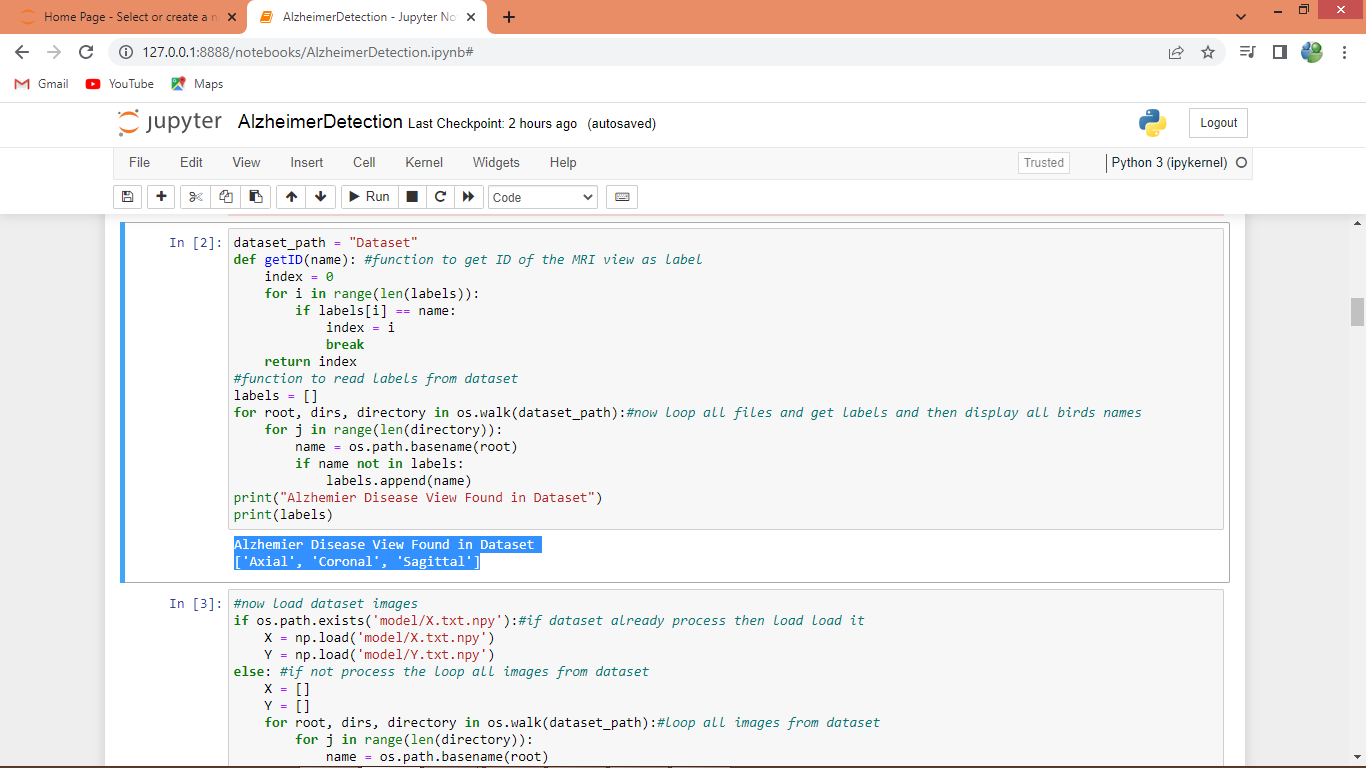
In propose work LENET was used with Maxpool2D and without any dropout layer so as extension we have modified LENET with extra layer called Maxpooling2D which is newer and consistent layer compare to existing Maxpool2D and can help in improving accuracy. In extension we have added Dropout layer to optimized filtered features.

SCREEN SHOTS

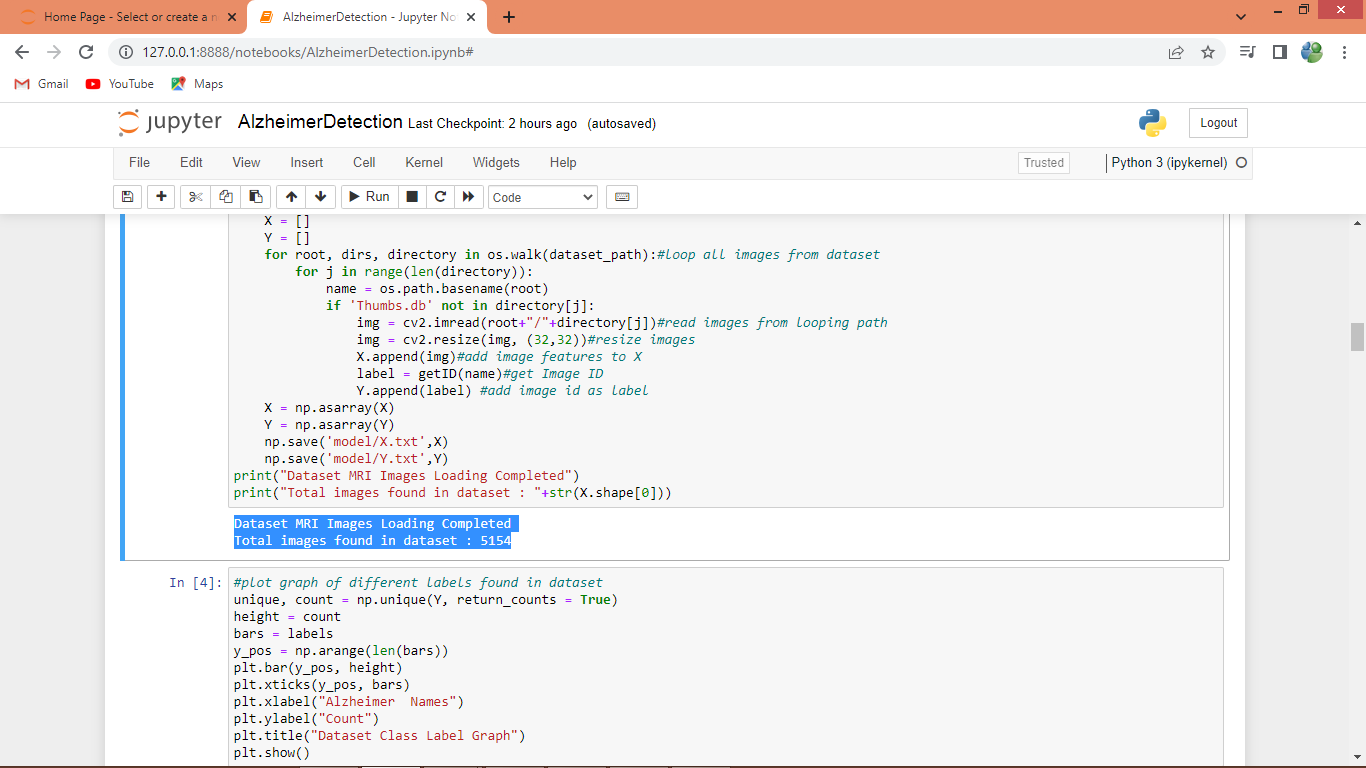
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



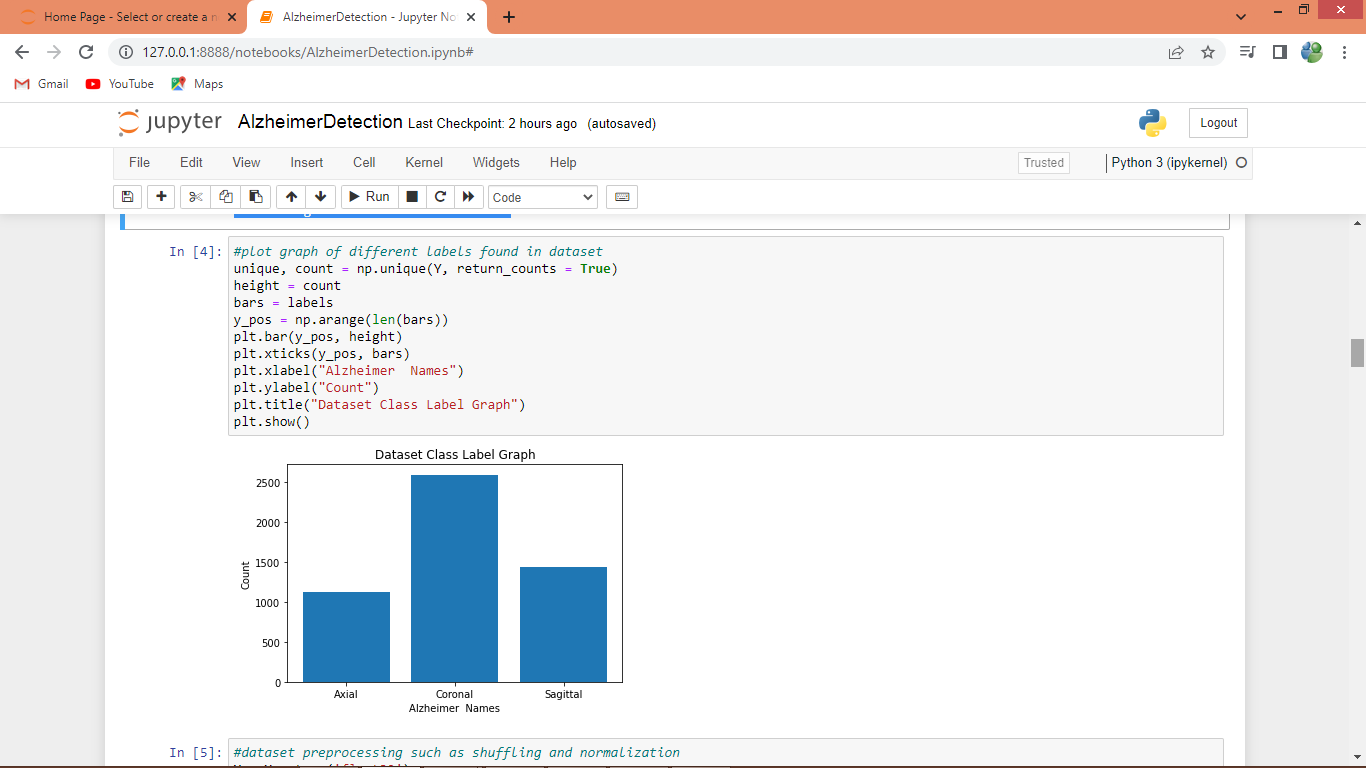
In above screen importing require python classes and packages



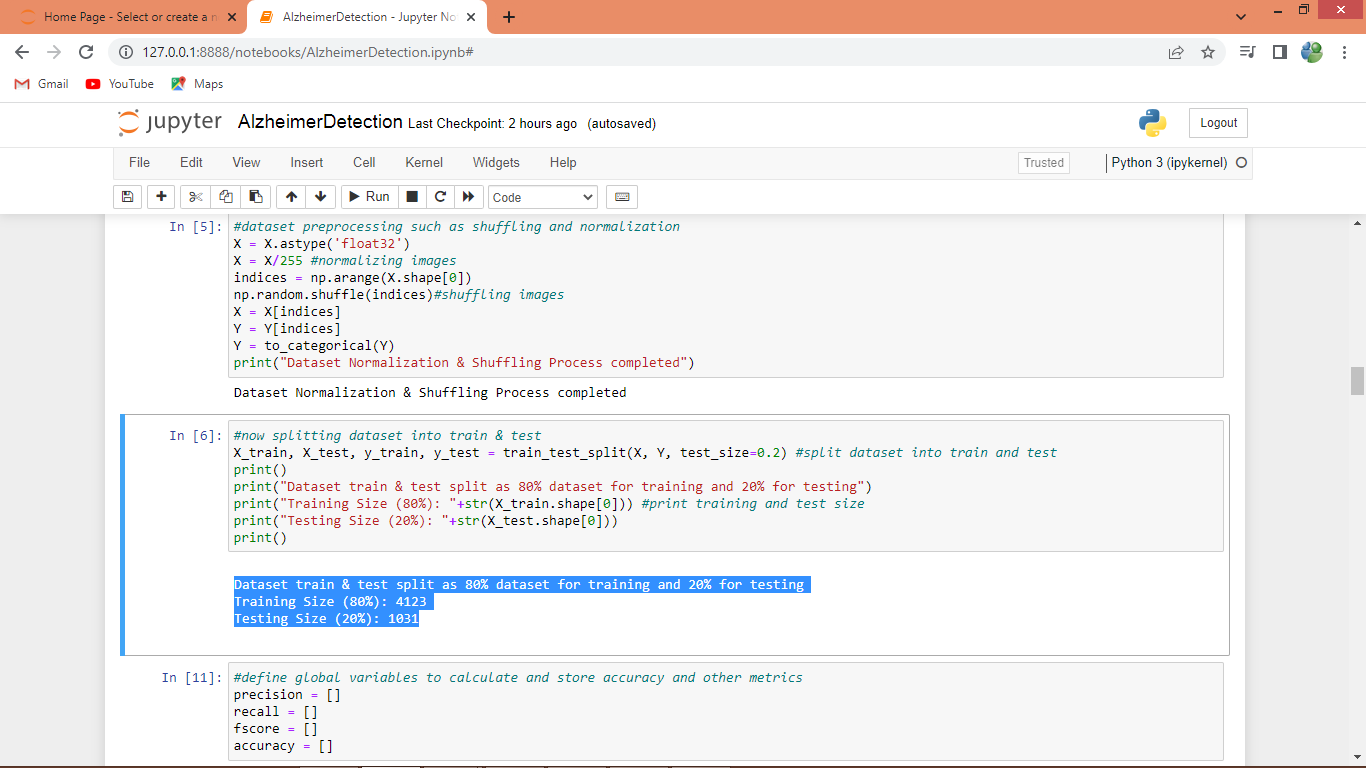
In above screen defining function to get labels from the dataset and in blue colour output displaying all labels

****

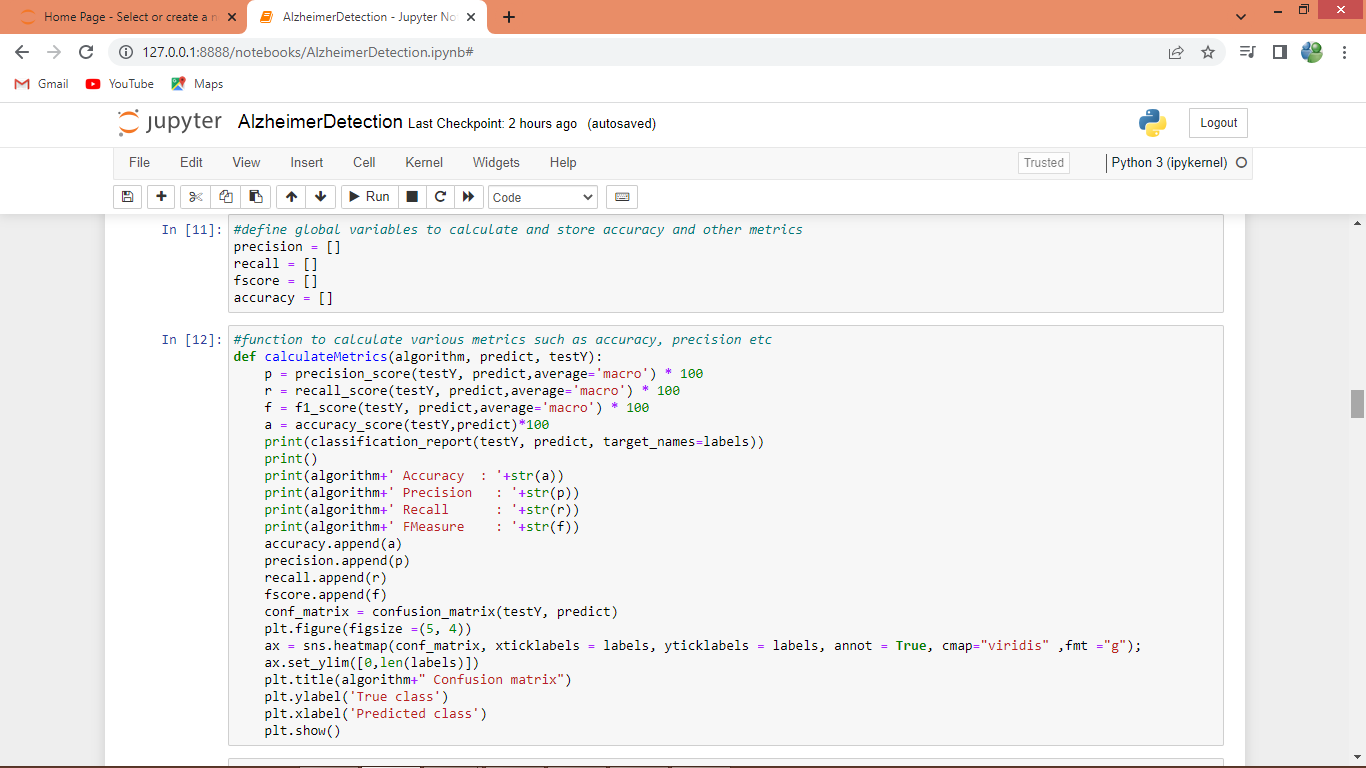
In above screen looping and reading all images from dataset and then displaying all loaded images in blue colour text



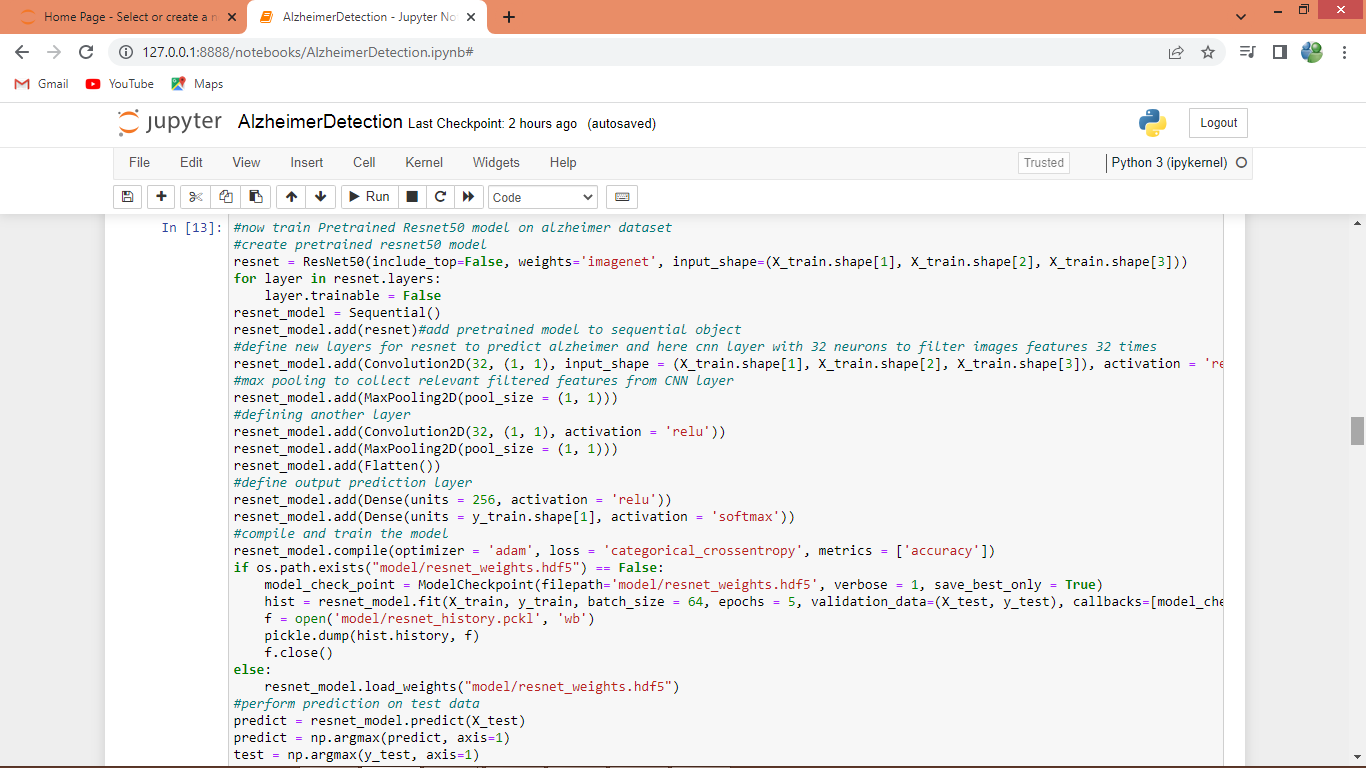
In above screen plotting graph of various labels found in dataset where x-axis represents label names and y-axis represents counts



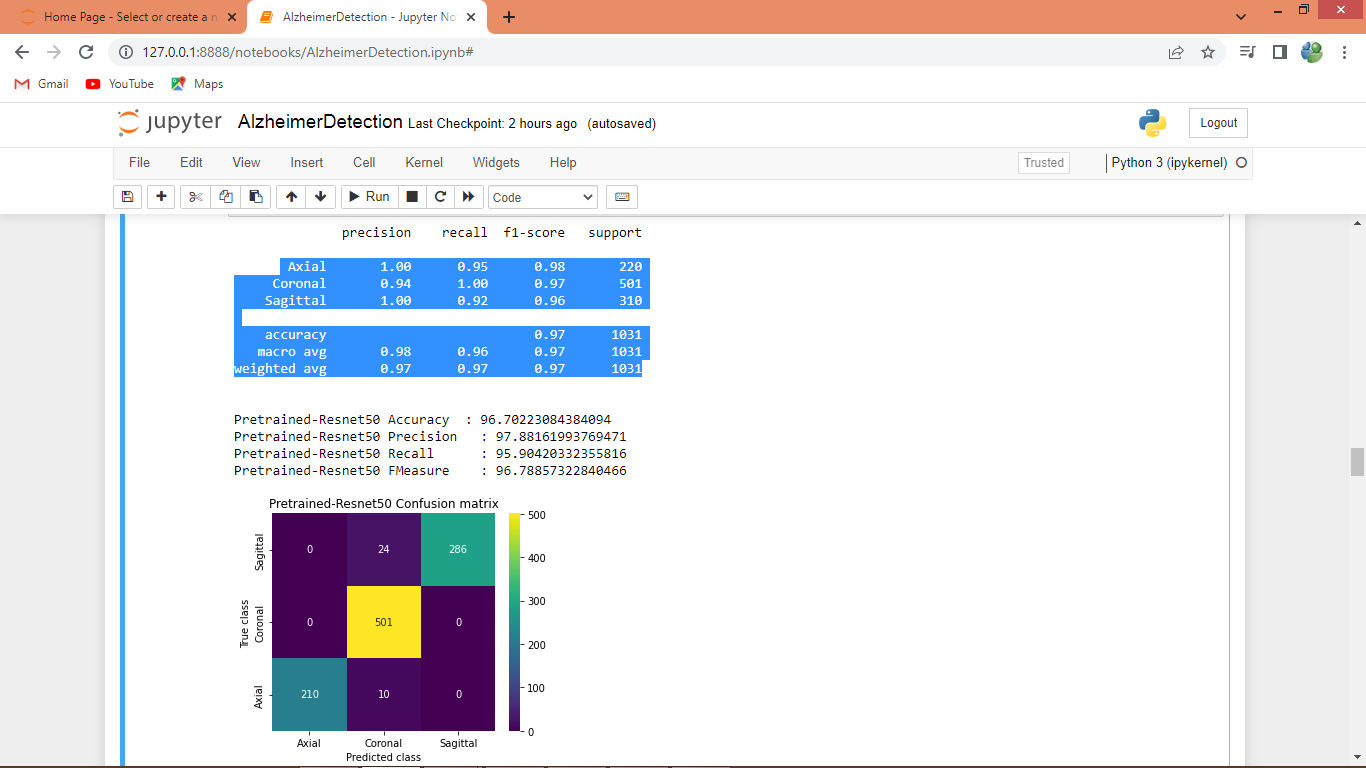
In above screen processing images features such as shuffling, normalization and splitting dataset into train and test where application using 80% images for training and 20% for testing



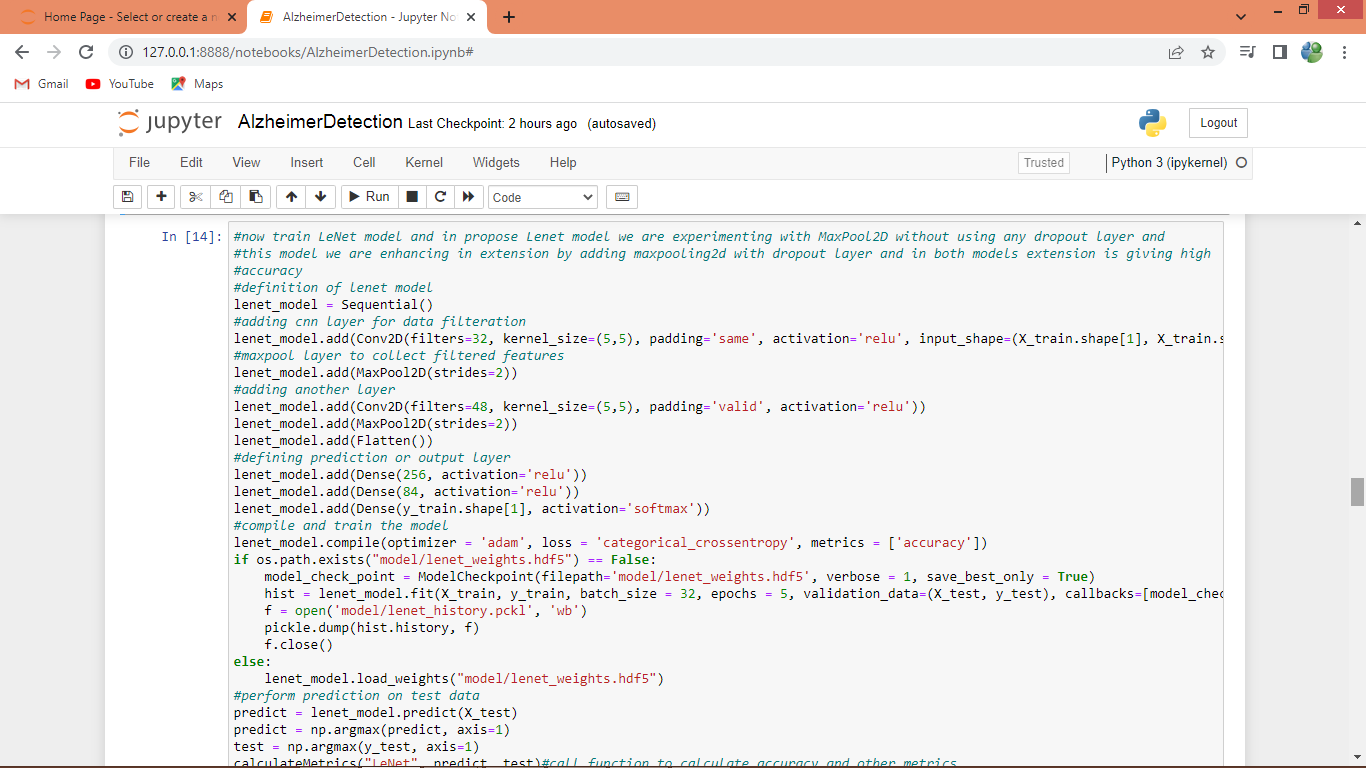
In above screen defining function to calculate accuracy and other metrics



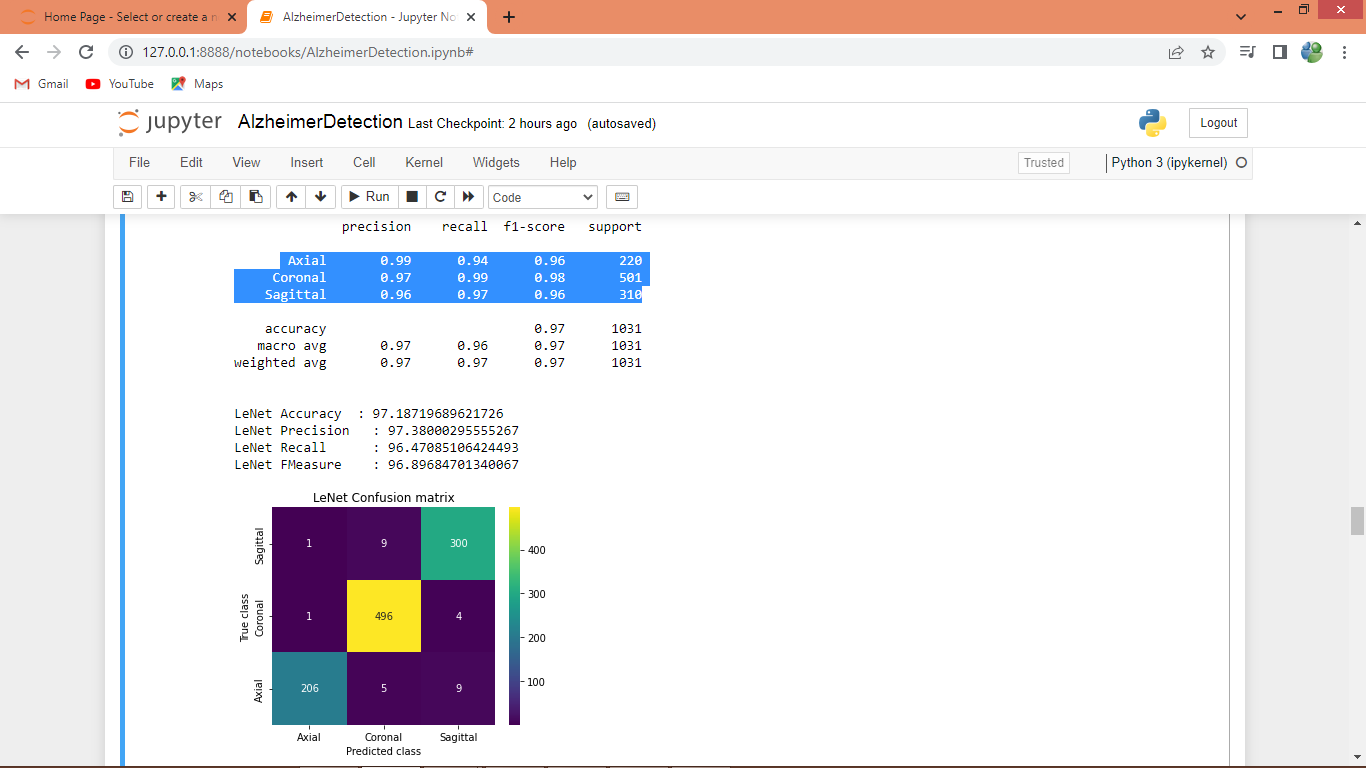
In above screen training pre-trained Resnet50 algorithm and after executing this block will get below output



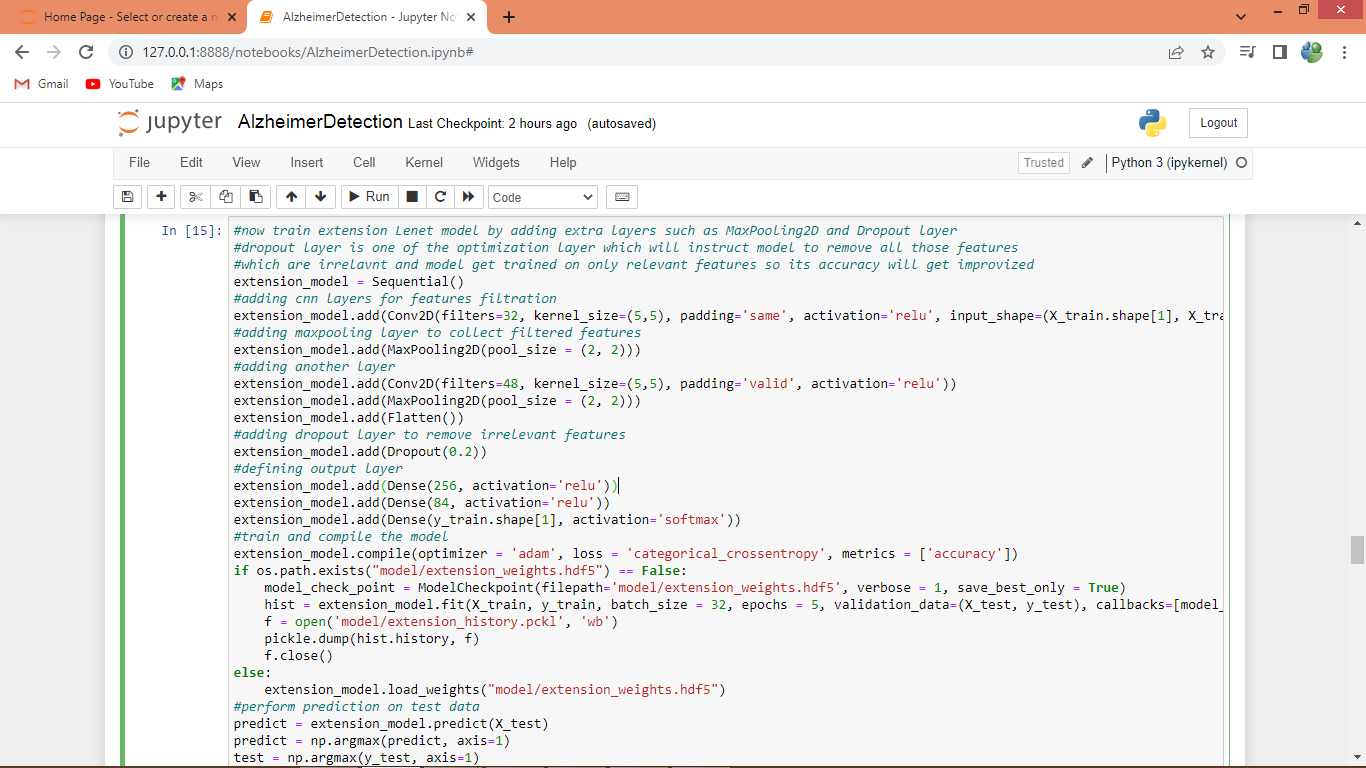
In above screen displaying Resnet50 output where blue colour text displaying accuracy, precision, recall and FSCORE from individual classes and then displaying overall accuracy as 96% and then we can see other metrics overall values and then in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels where different colour boxes represents correct prediction count and all blue boxes represents incorrect prediction count which are very few.



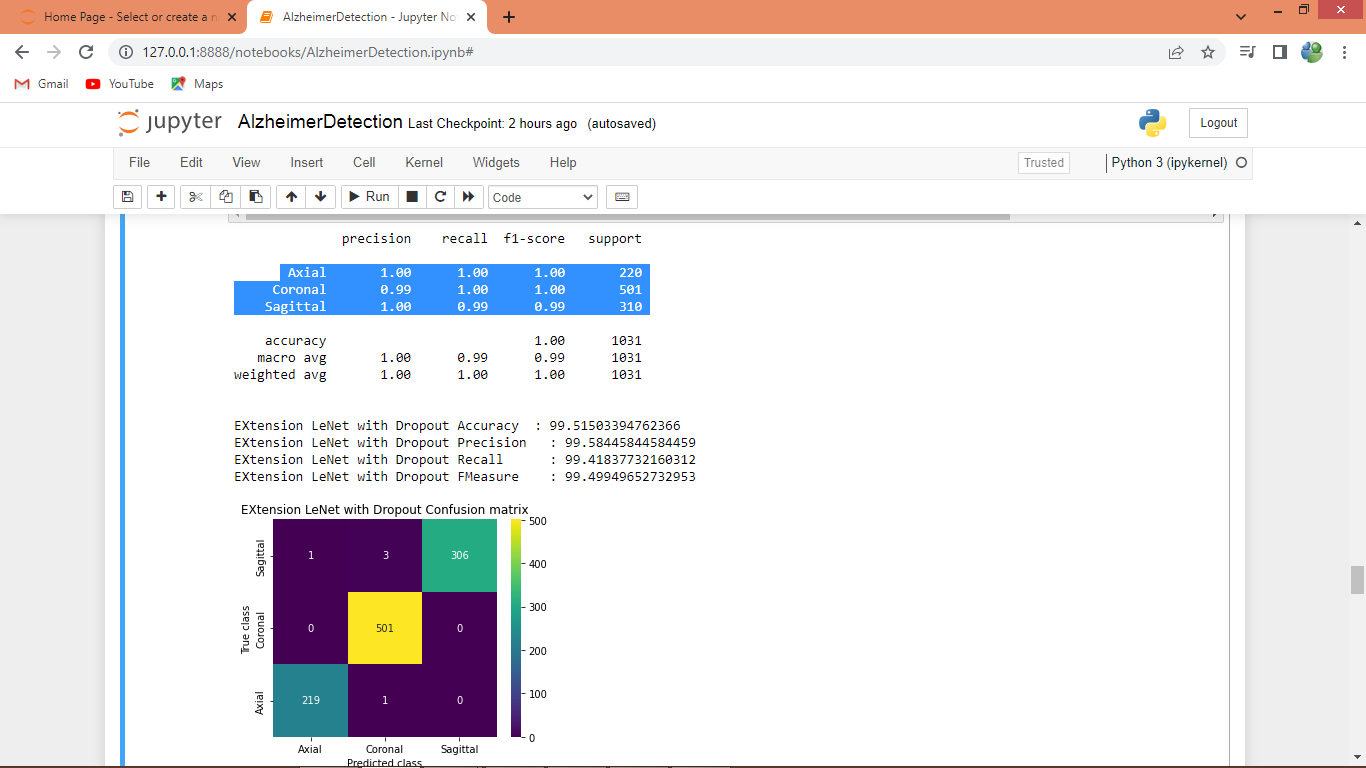
In above screen training LENET model and after executing this block will get below output



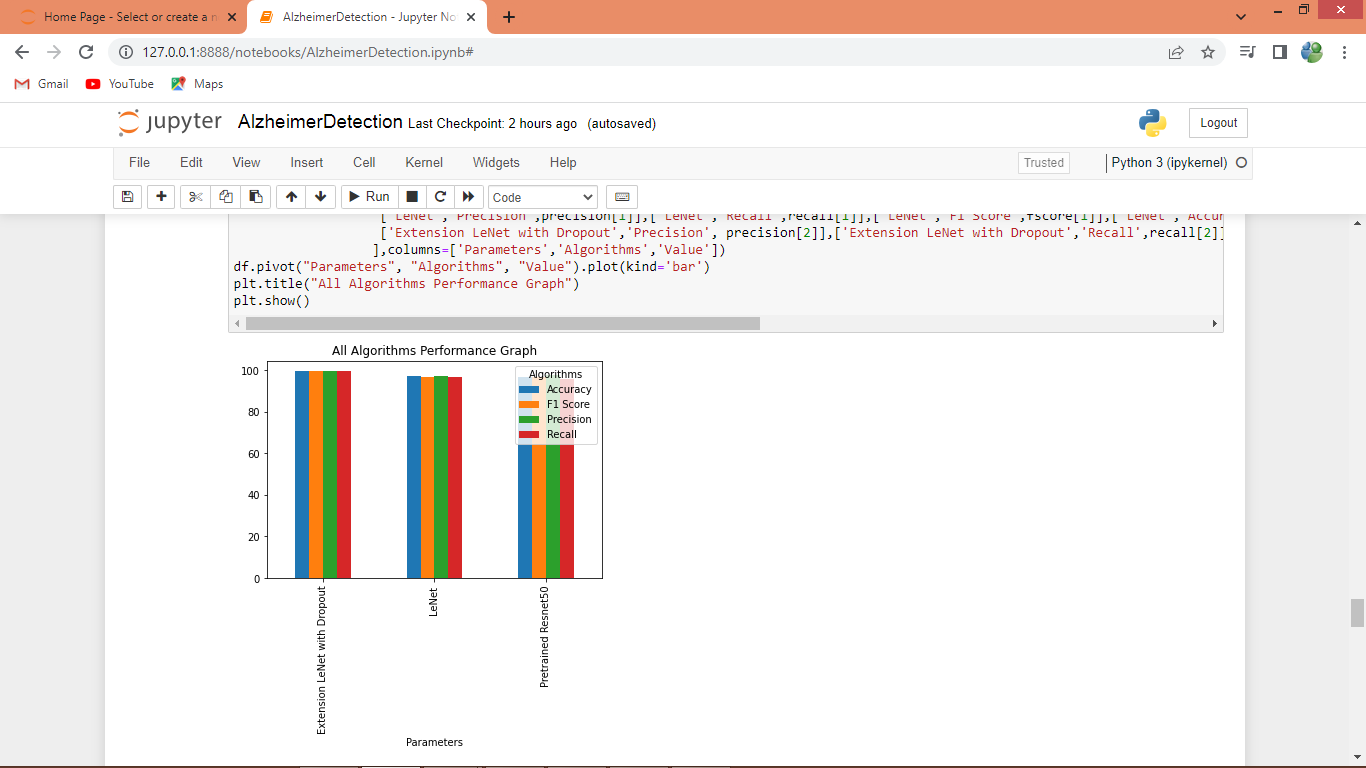
In above screen for LENET we can see individual class metrics and overall metrics where LENET overall accuracy is 97.19% accuracy and can see other metrics



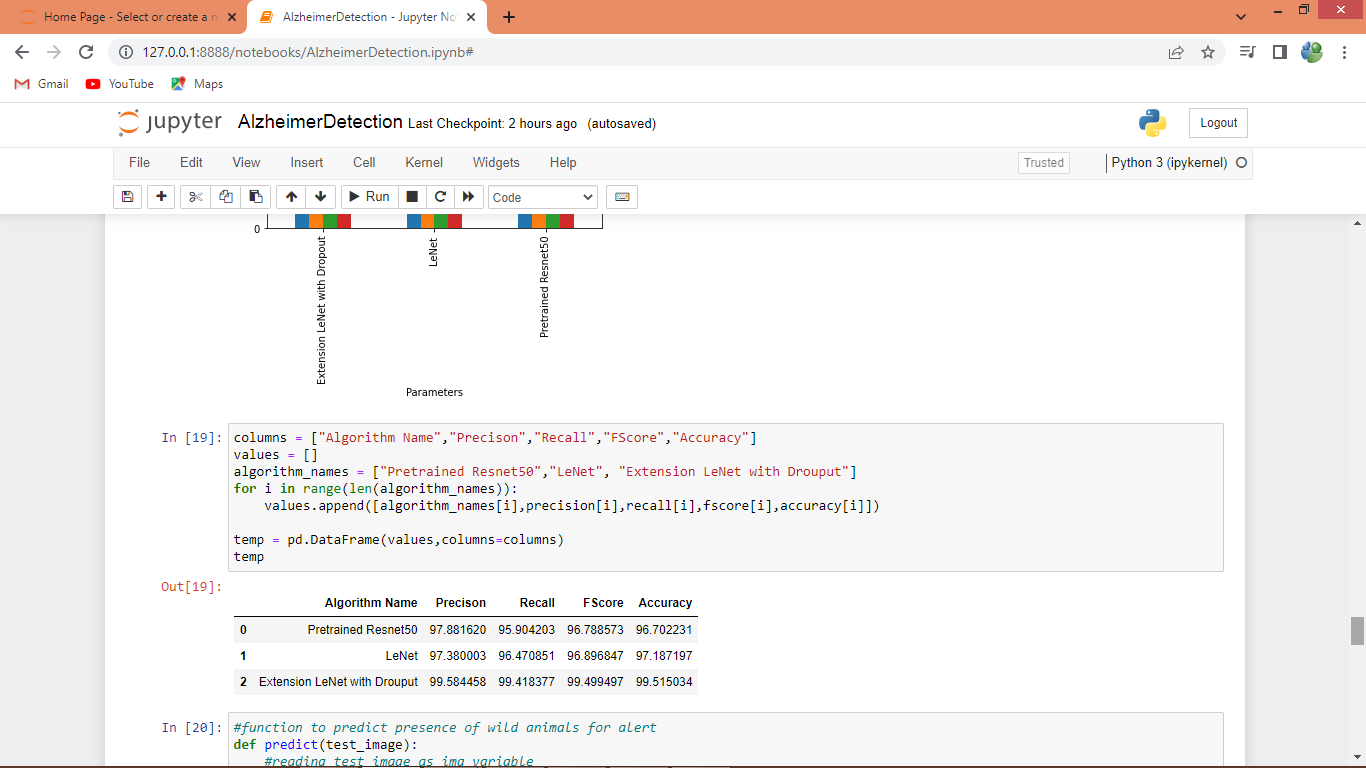
In above screen training LENET with extra layers as extension and after executing this block will get below output



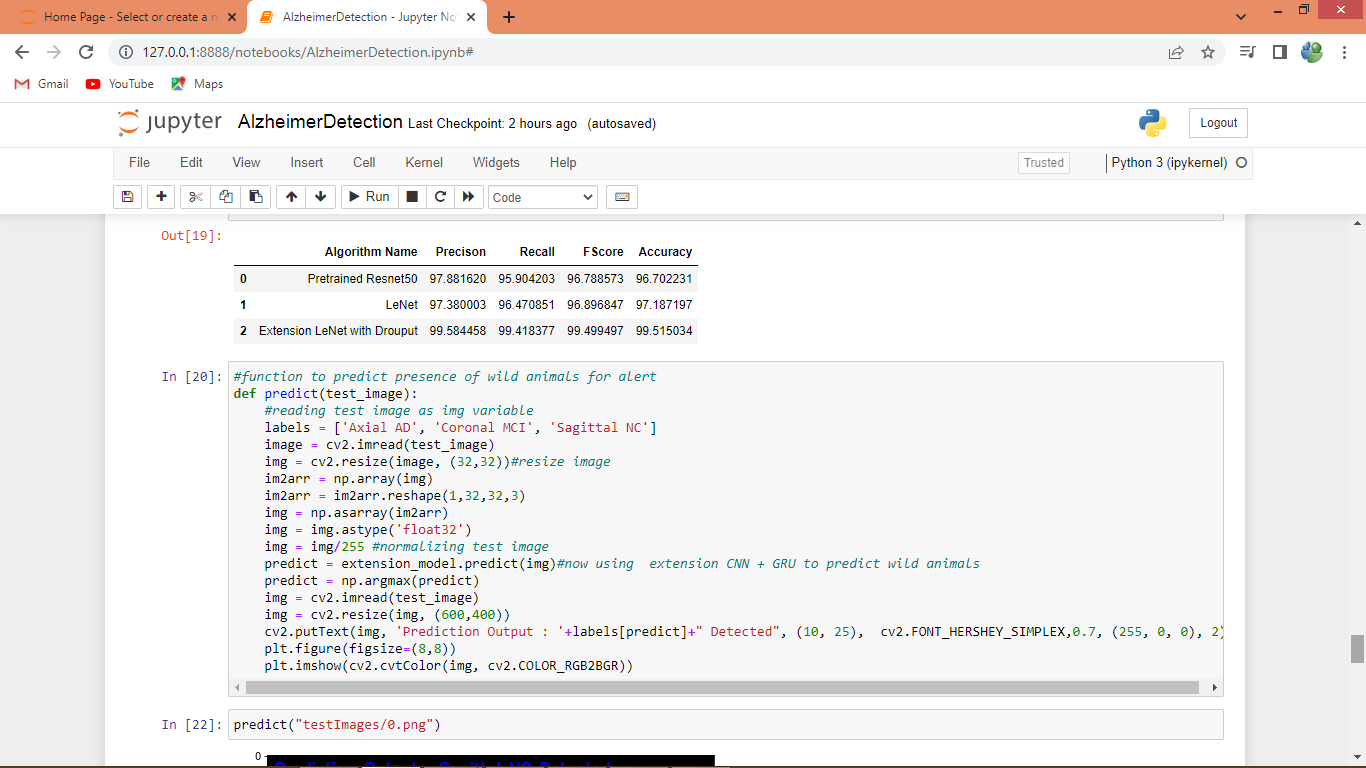
In above screen extension model got overall accuracy as 99.51% which is higher than other algorithms



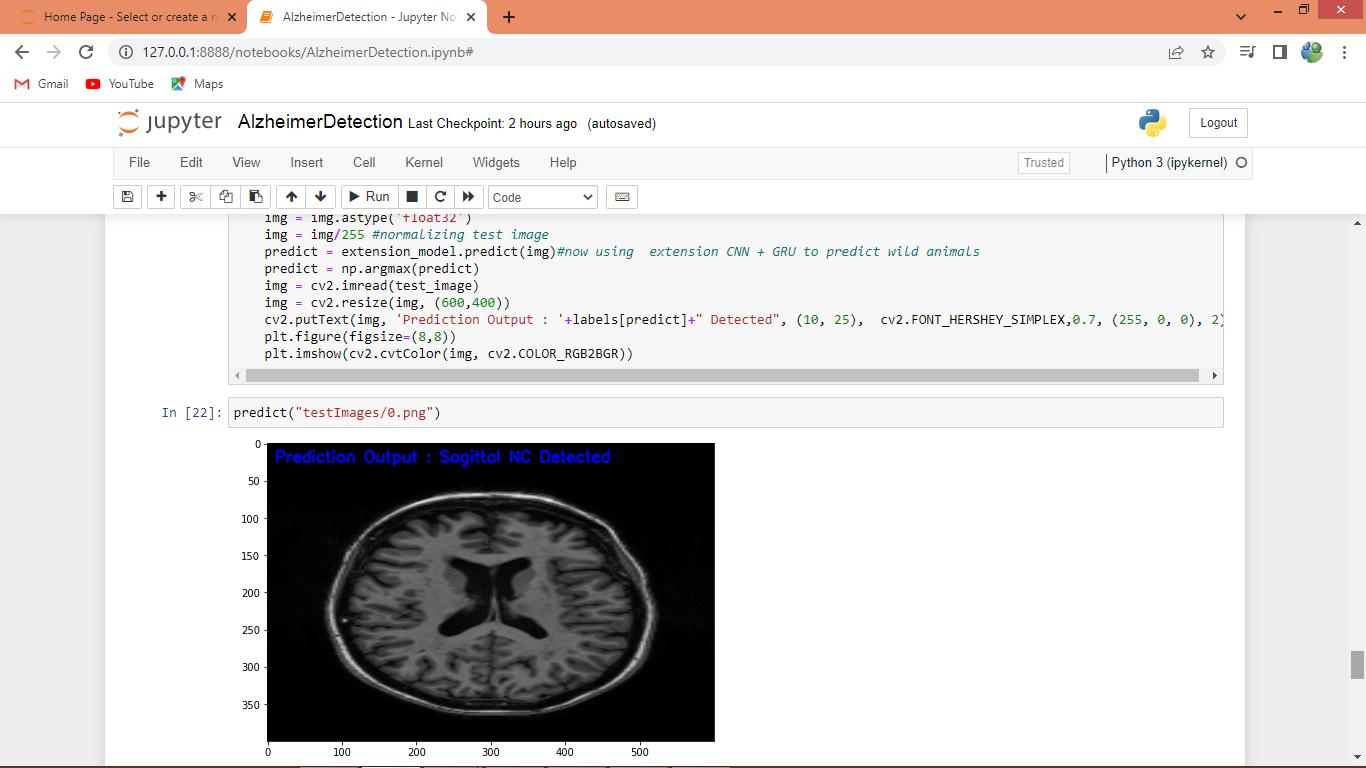
In above graph displaying performance of all algorithms in graph format where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms extension got high performance



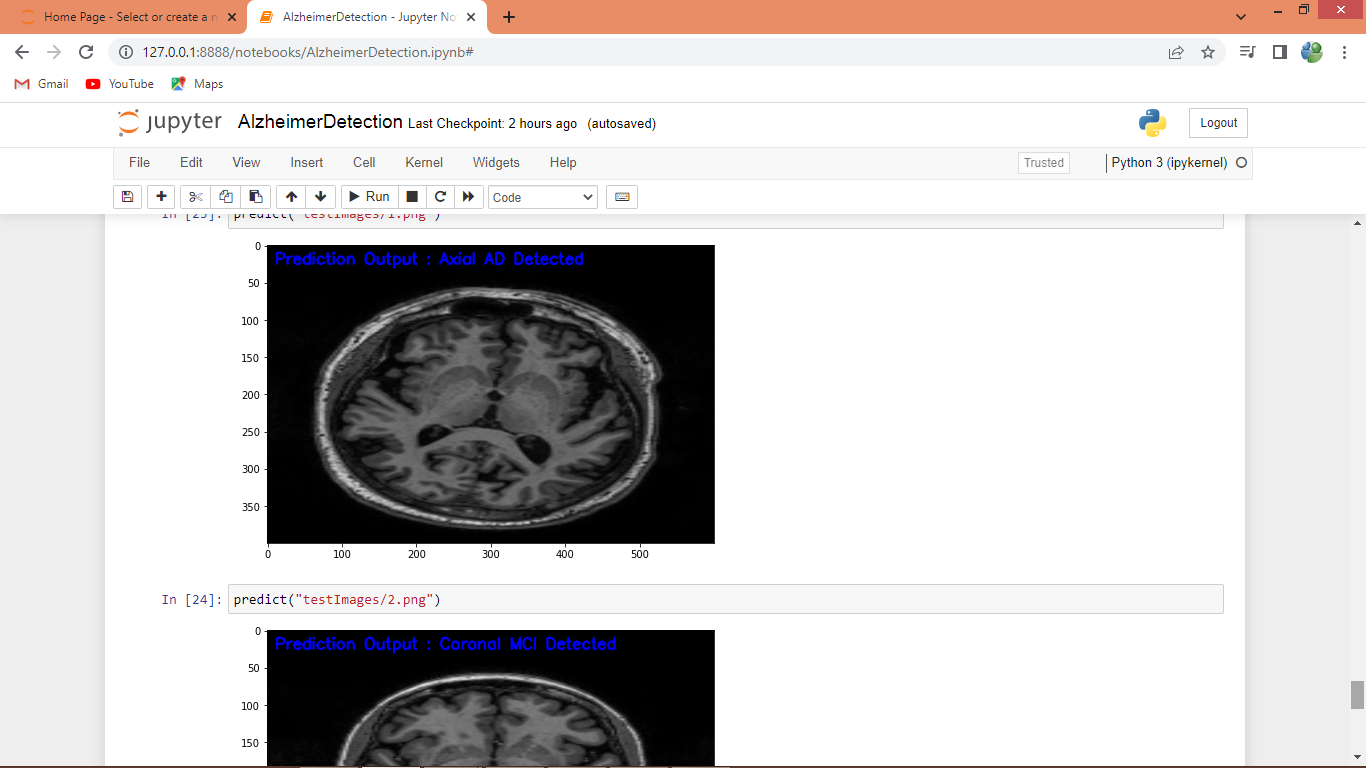
In above screen displaying all algorithm performance in tabular format



In above screen defining predict function to read test images and then normalize and predict disease using extension model



In above screen given test image is predicted as ‘Sagittal NC’



In above screen we can see predicted output displaying in blue colour text.