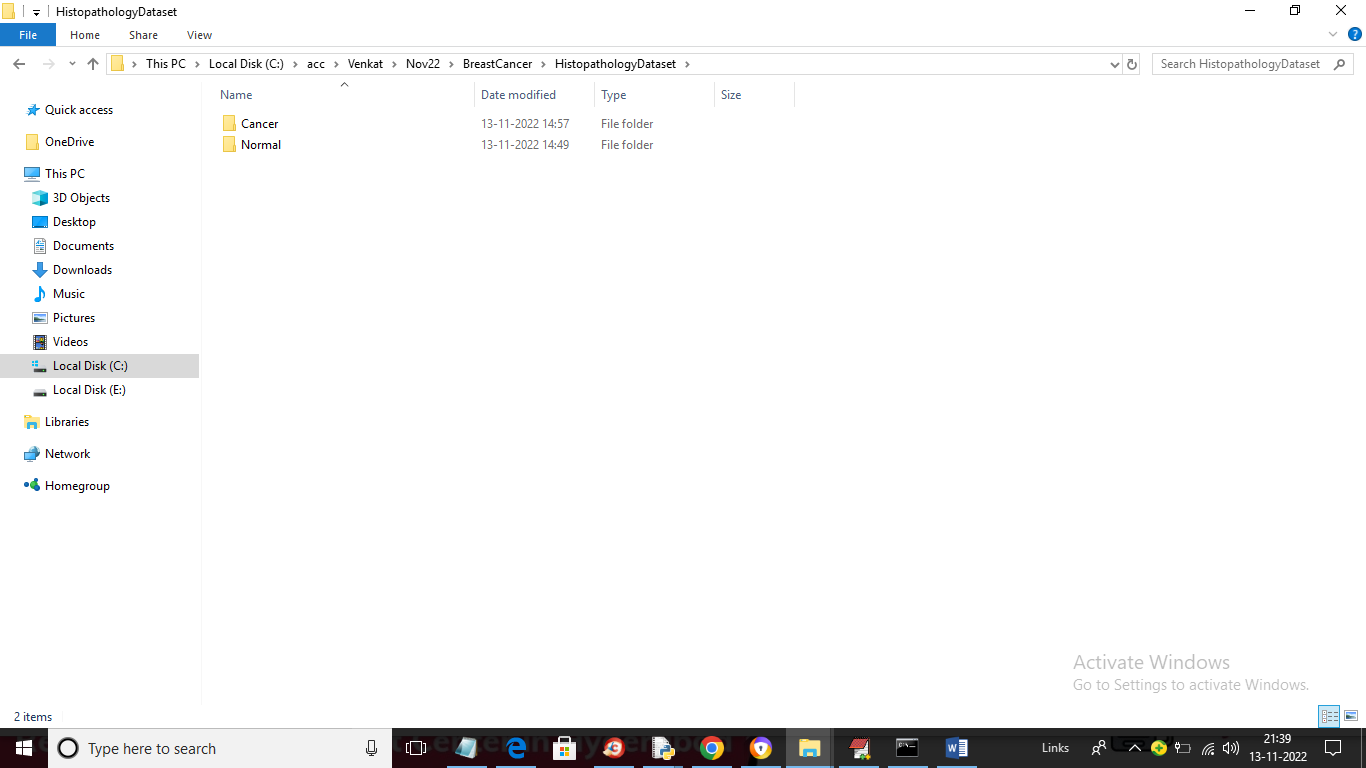
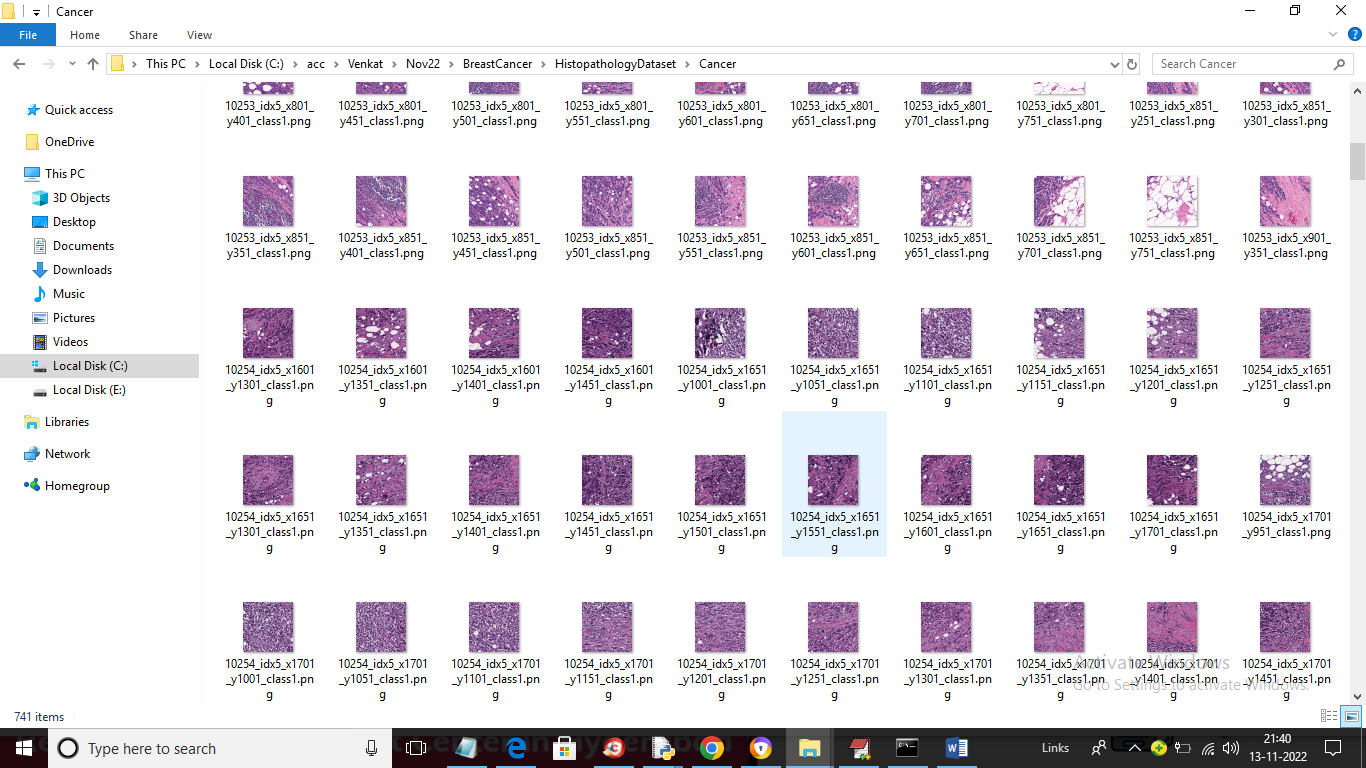
Breast Cancer Diagnosis on Pathological Images Data Augmentation Method: Cycle GAN

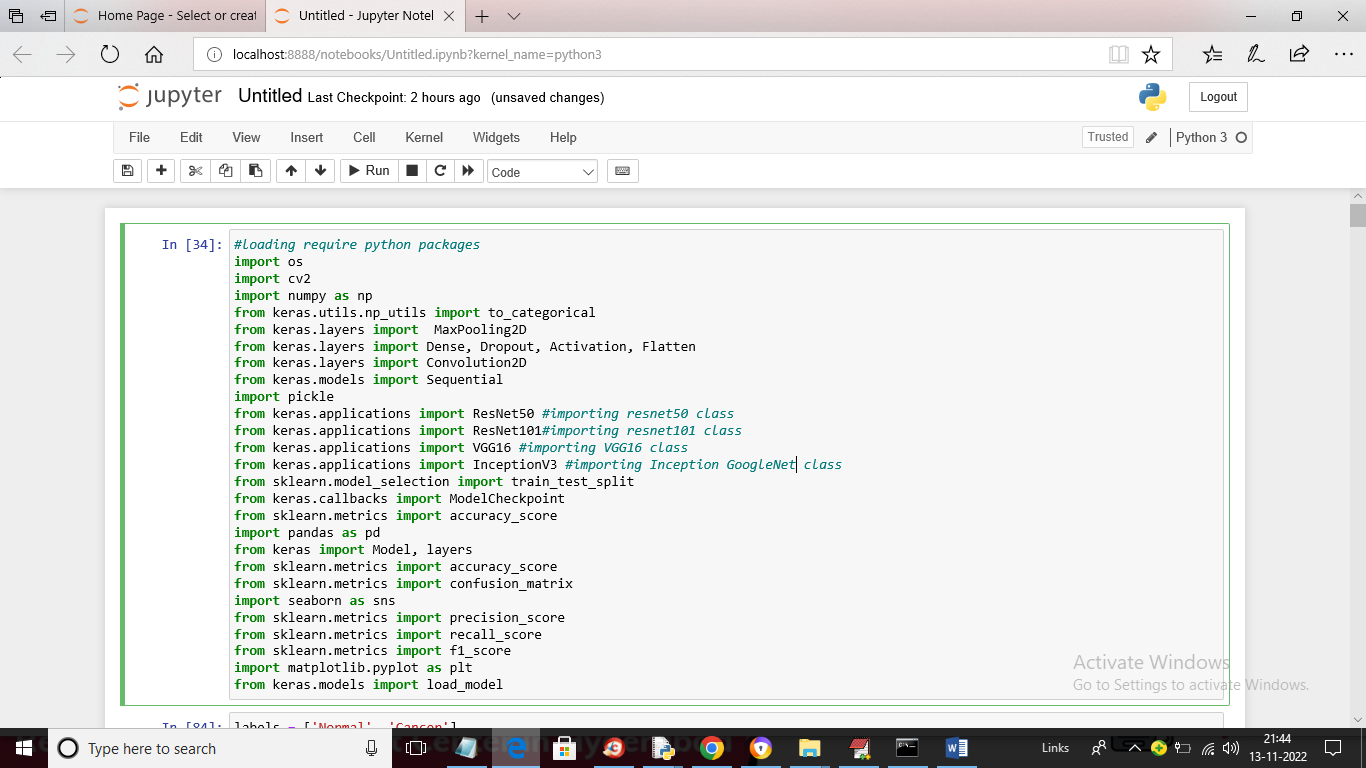
In this project we are using Histopathology images to predict breast cell as Normal or Cancer and for prediction we are experimenting with various Deep Learning algorithms such as Resnet50, Resnet101, GoogleNet, VGG16 and AlexNet and among all algorithms VGG16 and AlexNet is giving best performance with an accuracy of more than 95%. In below screen we are showing dataset images used to train above discussed algorithms



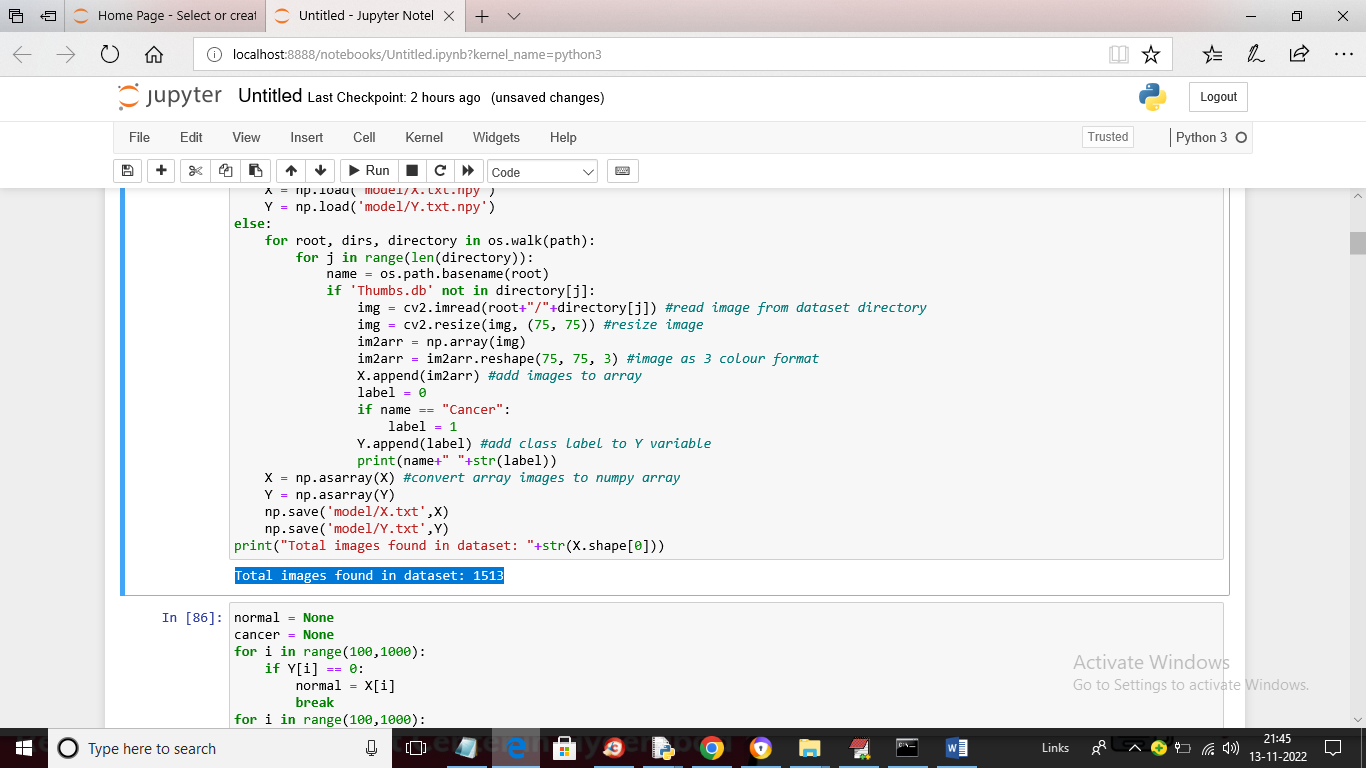
In above screen we can see dataset contains 2 folders called Normal and Cancer and just go inside any folder to view images



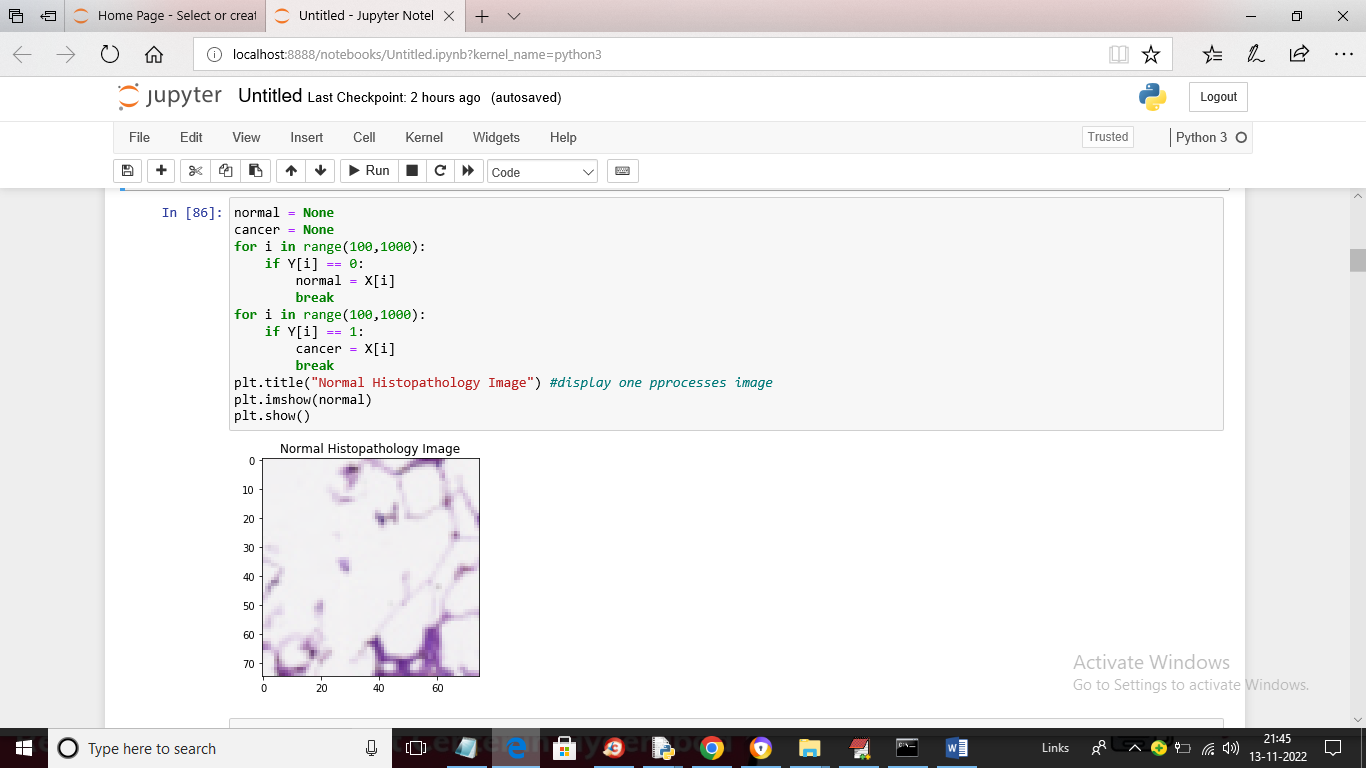
So by using above images we are training all the algorithms and to train this algorithms we have used JUPYTER notebook and below screen showing code and output details. Each block in JUPYTER designed for specific purpose and you can read blue colour comments to know about the purpose



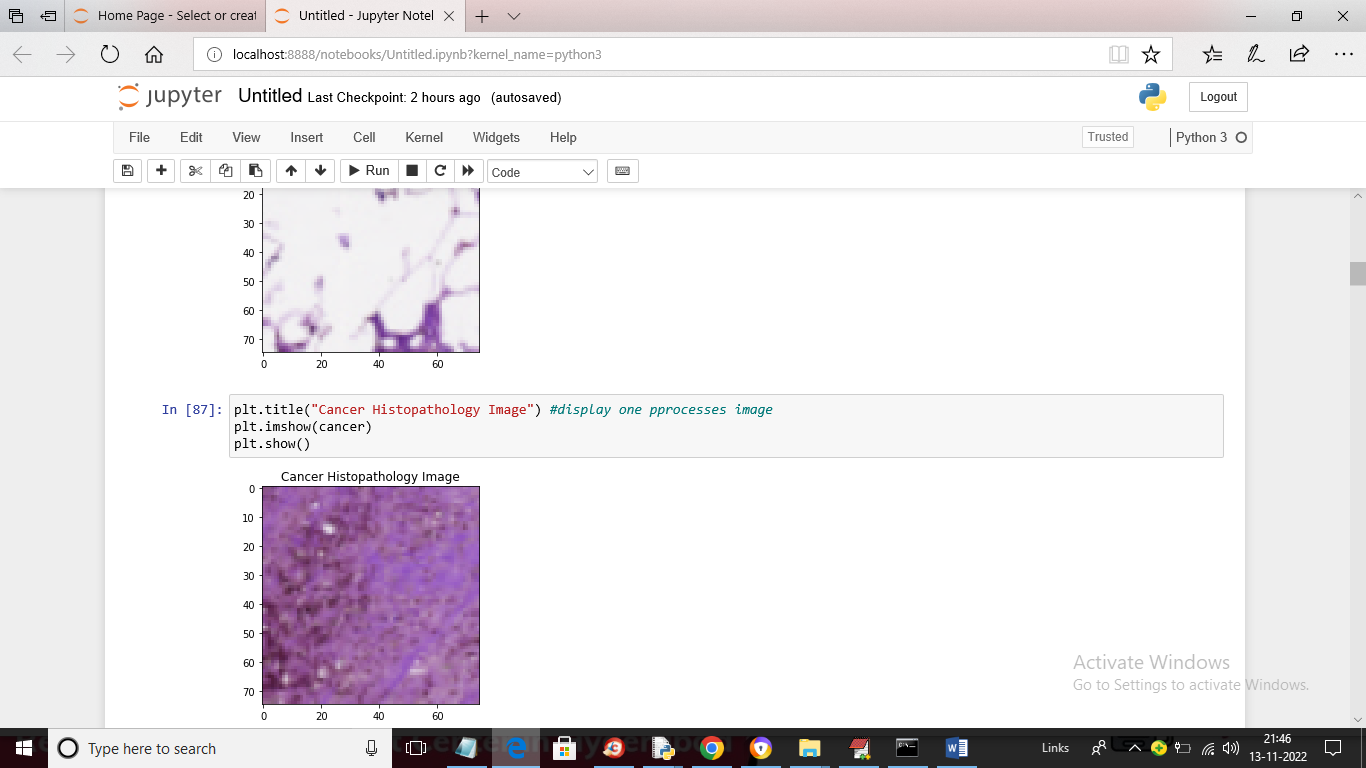
In above screen we are loading require python packages



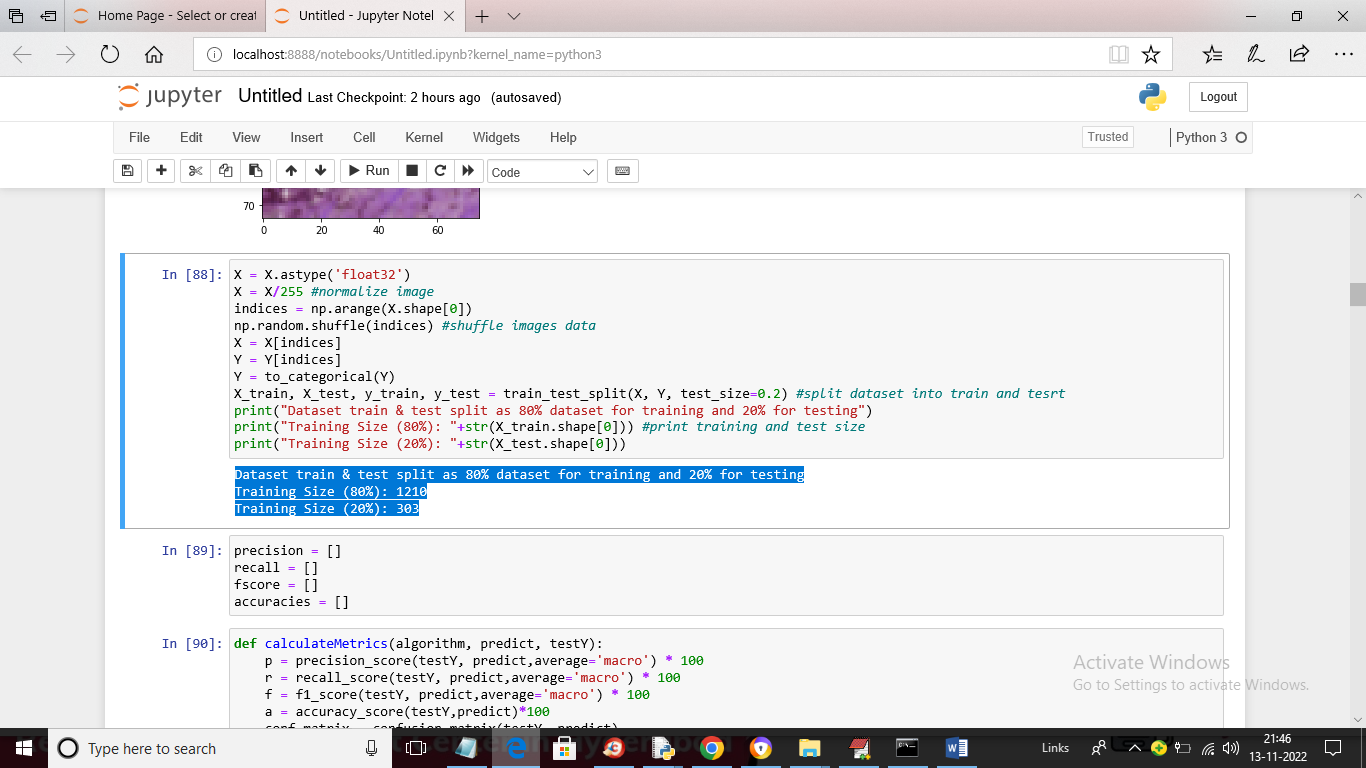
In above screen we are reading images from dataset and then adding images and X array and class label to Y array and in blue colour text we can see dataset contains more than 1500 Images



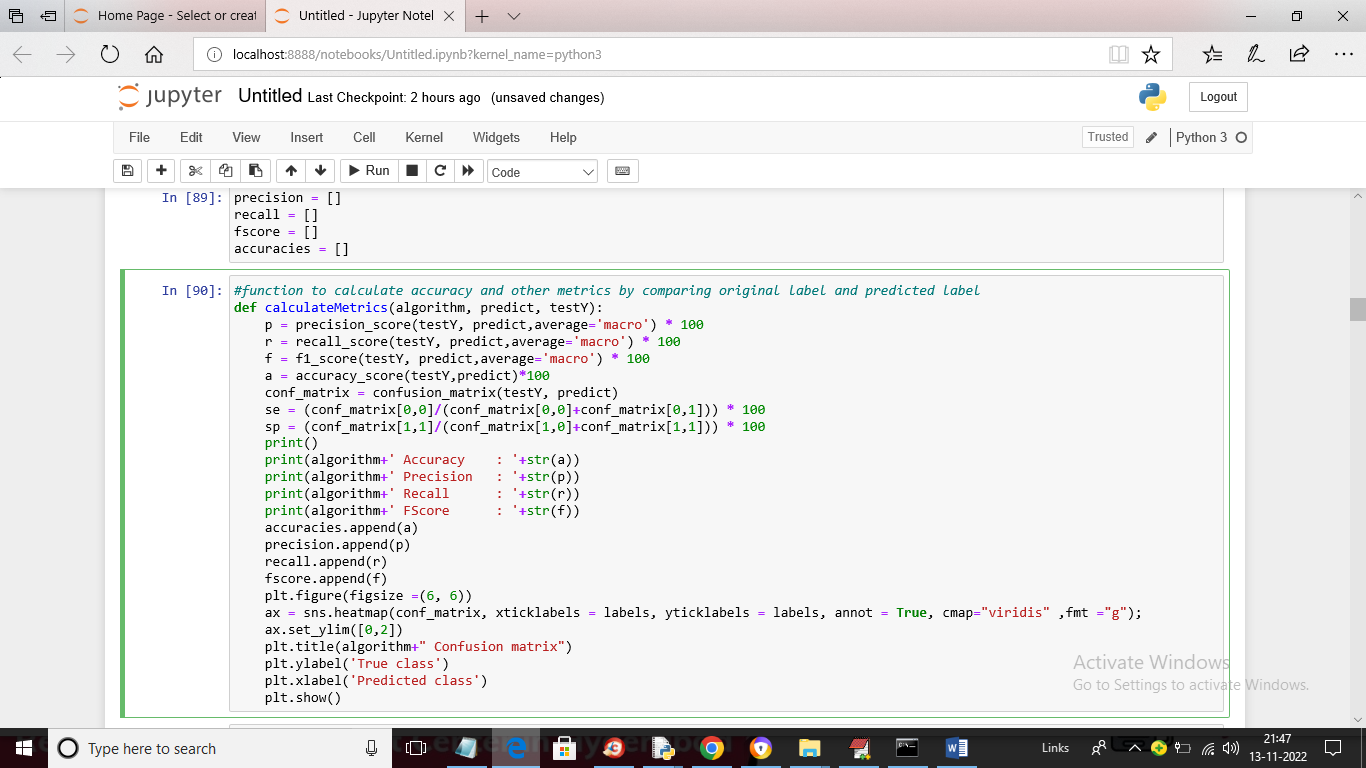
In above screen we are displaying processed Breast Cell image



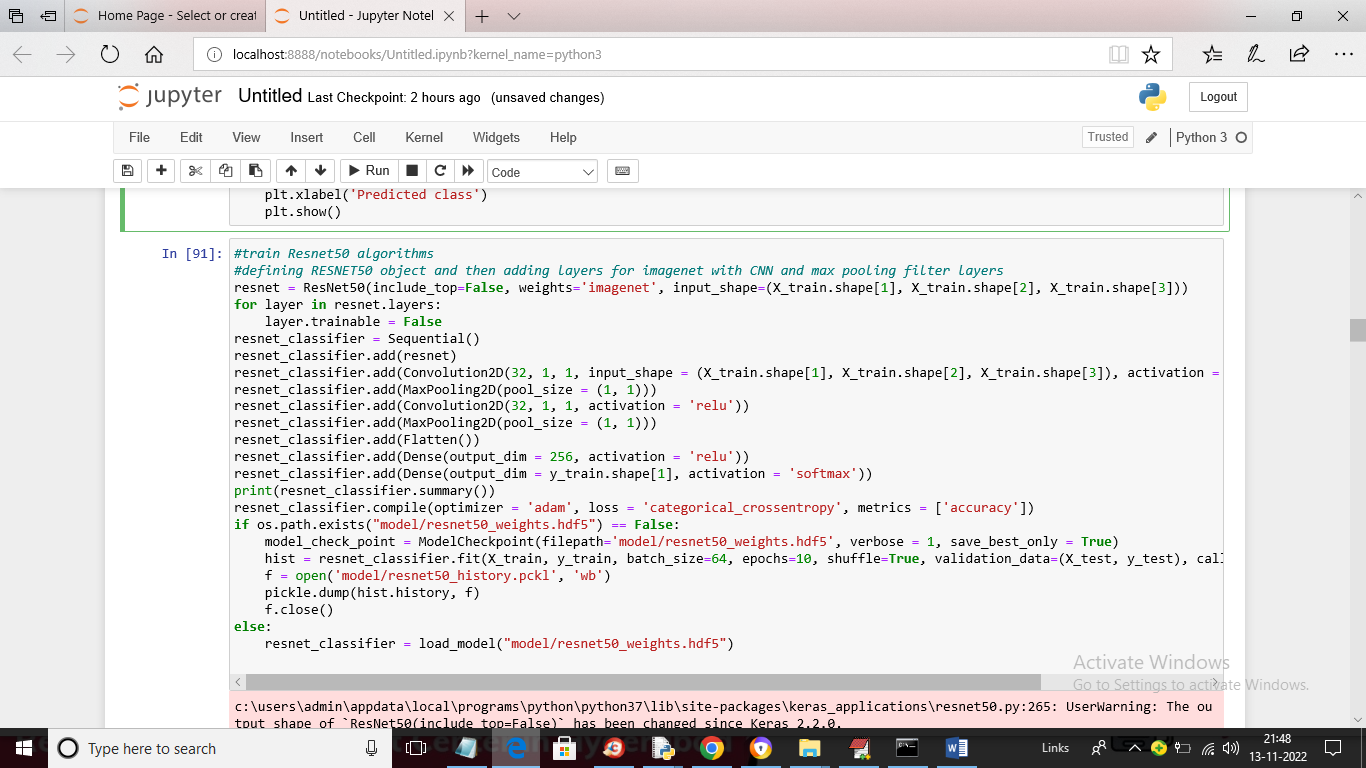
In above screen we are displaying sample processed Cancer image



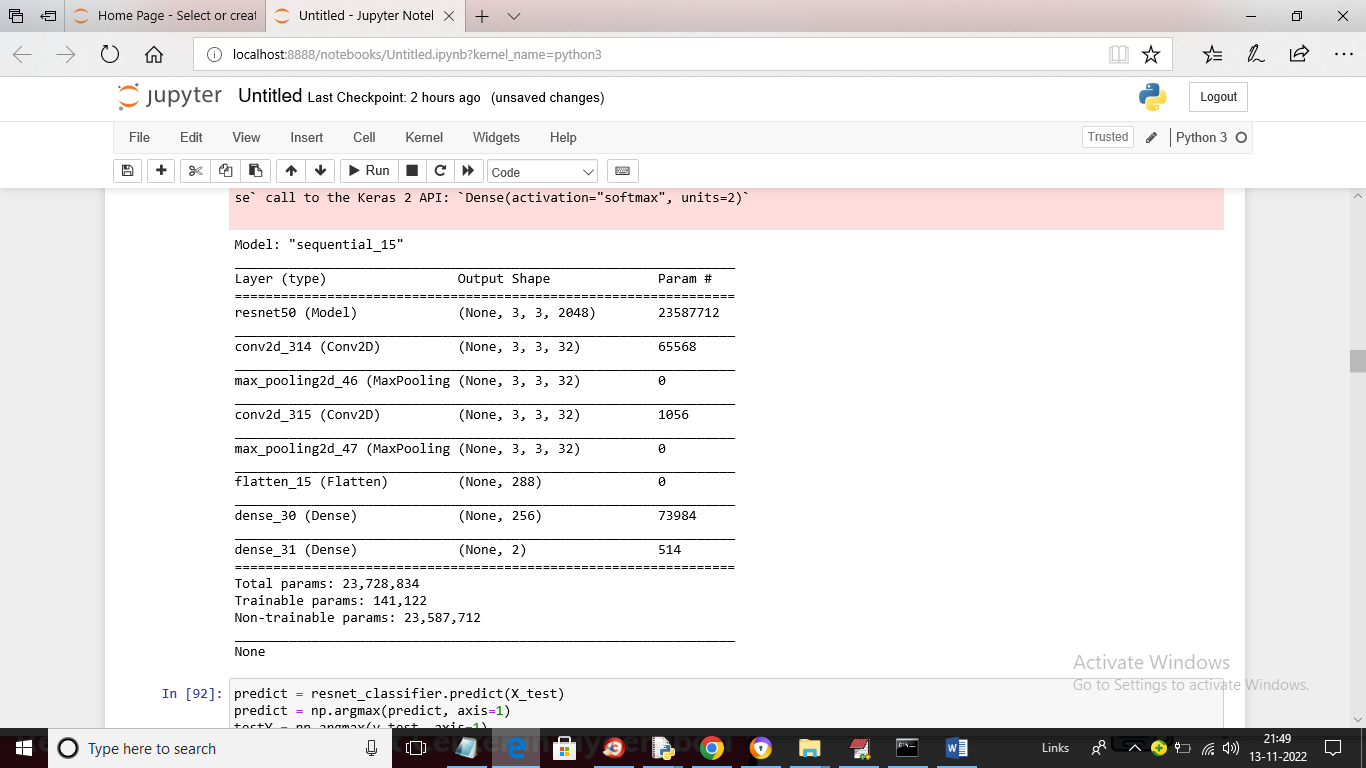
In above screen we are normalizing each image and then shuffling images and then splitting dataset into train and where application used 80% images for training and 20% for testing and in blue colour text we can see total images used for training and testing



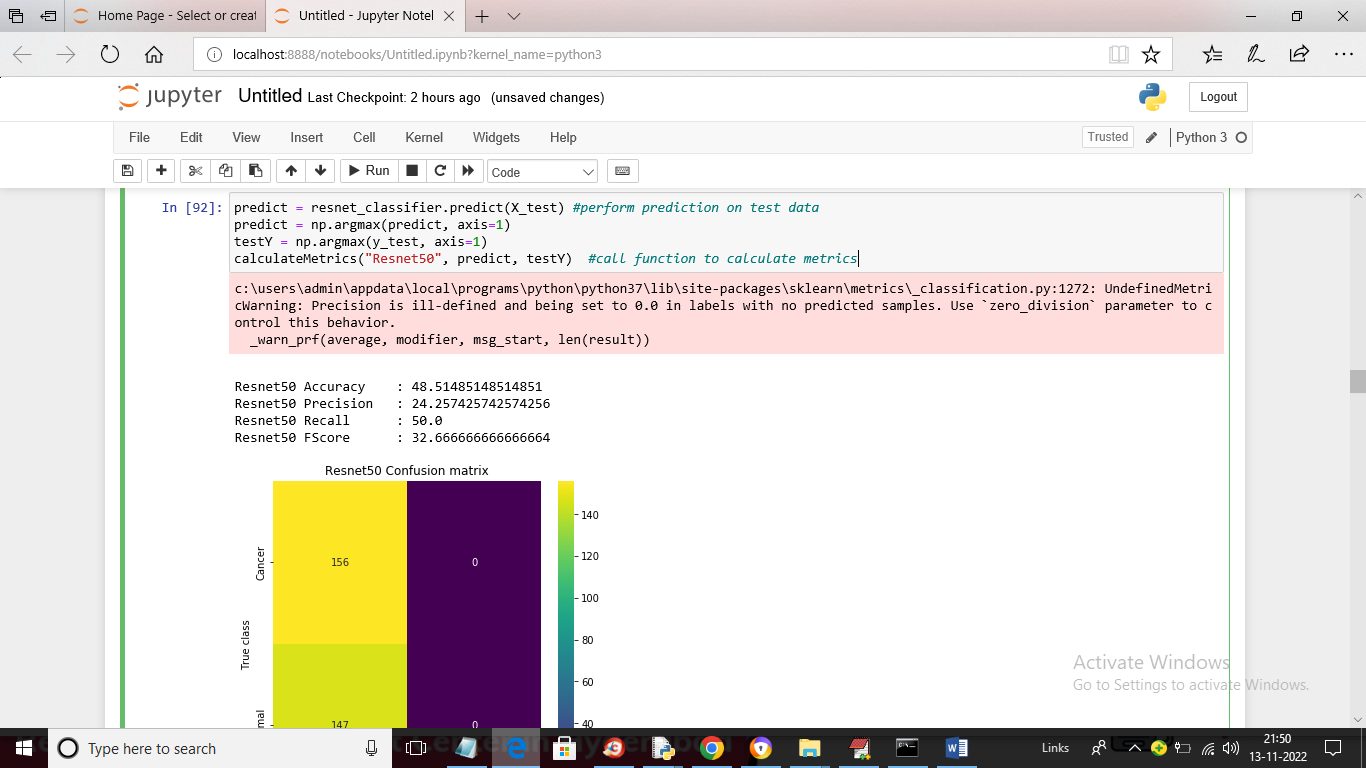
In above screen we have defined function to calculate accuracy and other metrics



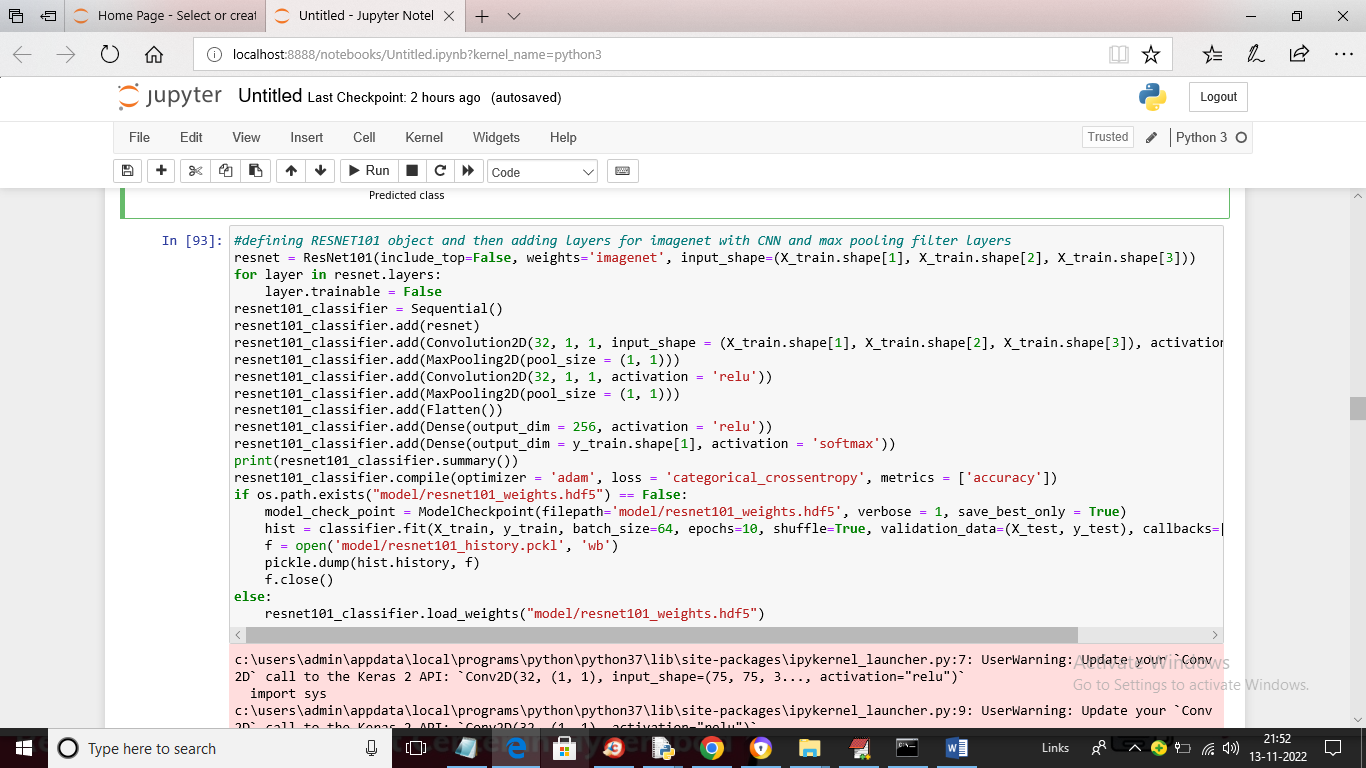
In above screen we are defining layers for RESNET50 and then training on train data and then validate on test data to build RESNET50 model and below is the architecture details of ResNet50



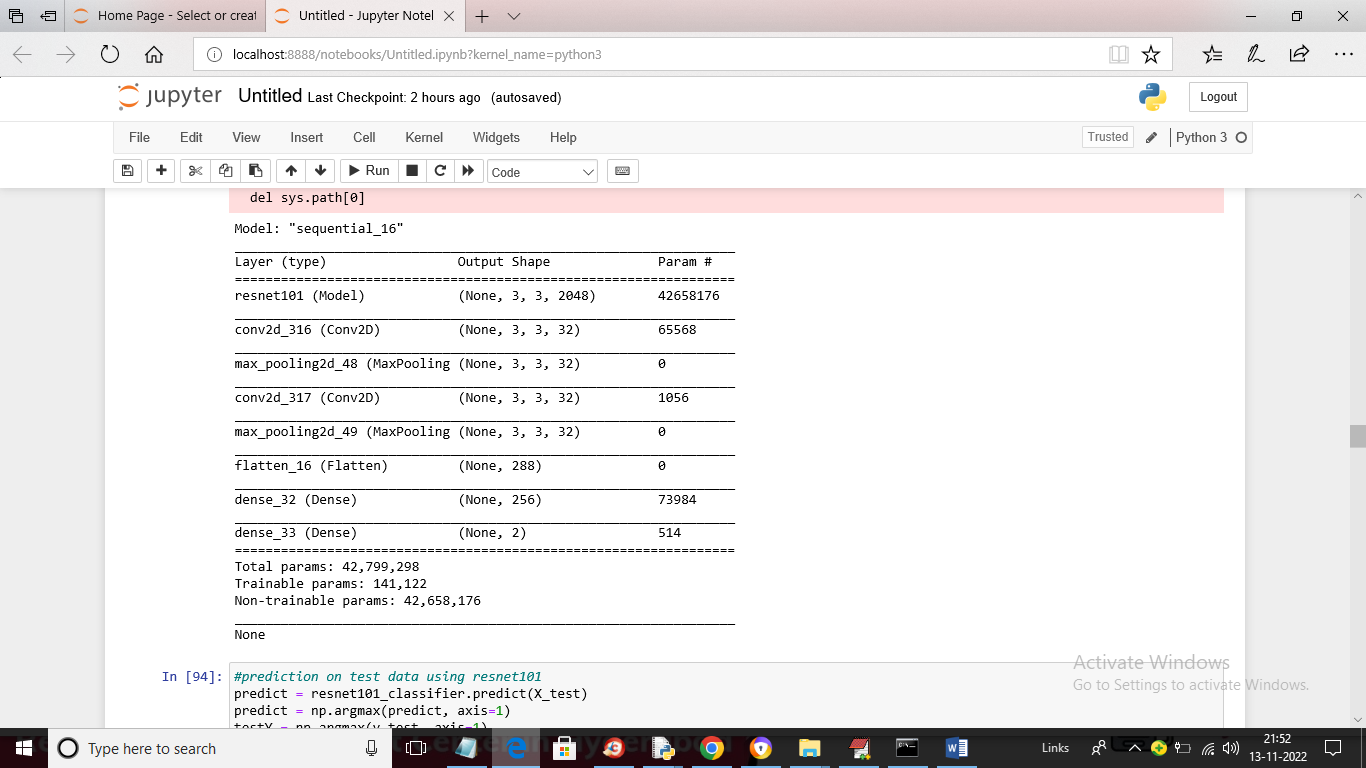
In above screen we can see Resnet50 layers and in below screen we are performing prediction on test data using Resnet50



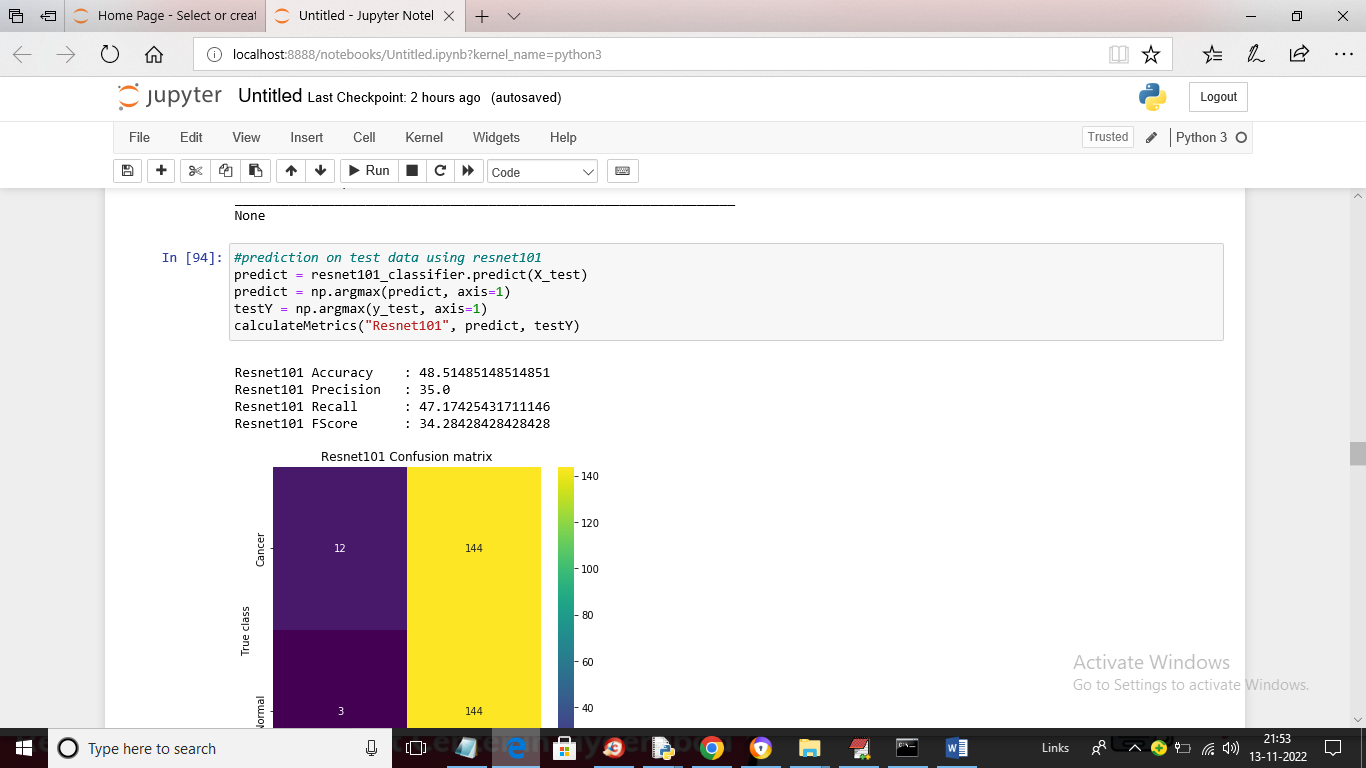
In above screen with Resnet50 we got 48% accuracy and in confusion matrix graph x-axis represents Predicted classed and y-axis represents True classes and the count in both x and y-axis for normal and cancer is the correct prediction count and other count are incorrect prediction.



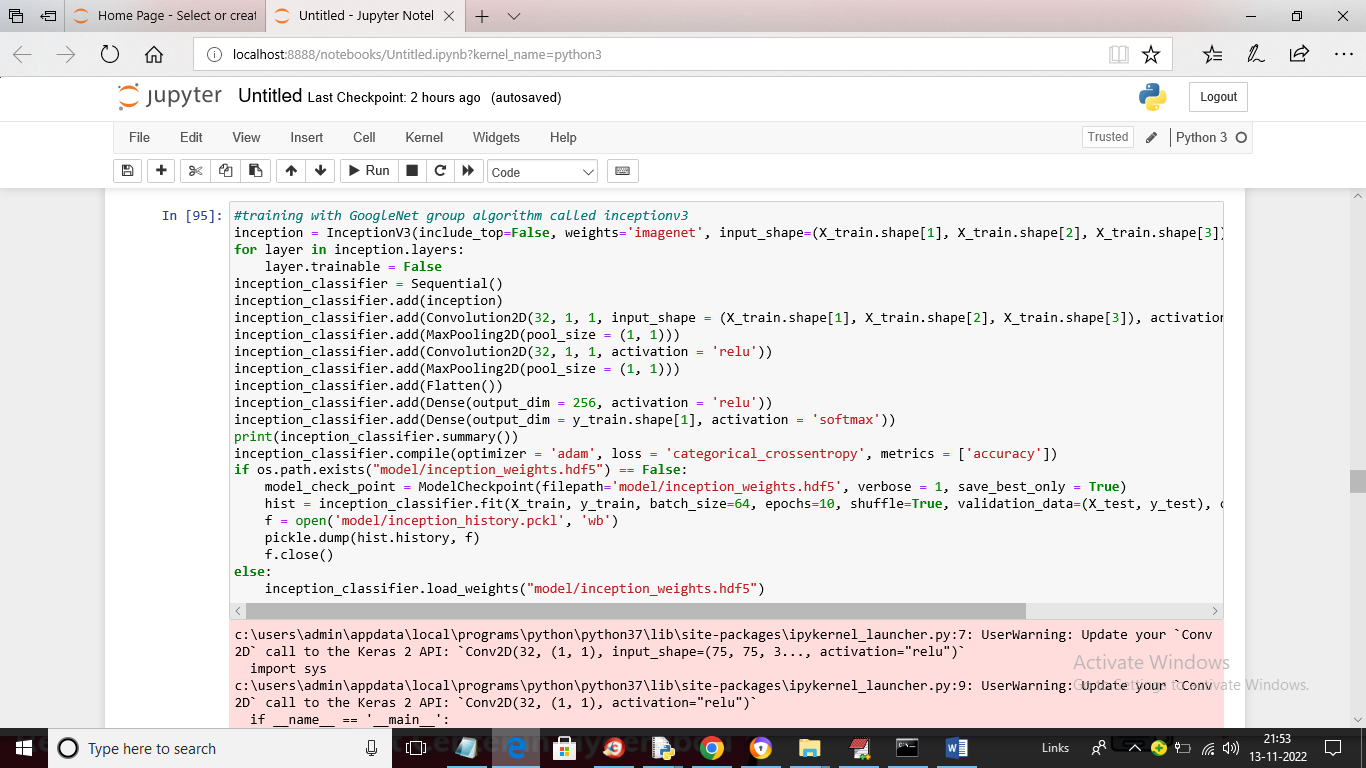
In above screen we are training with Resnet101 and below is the layer details



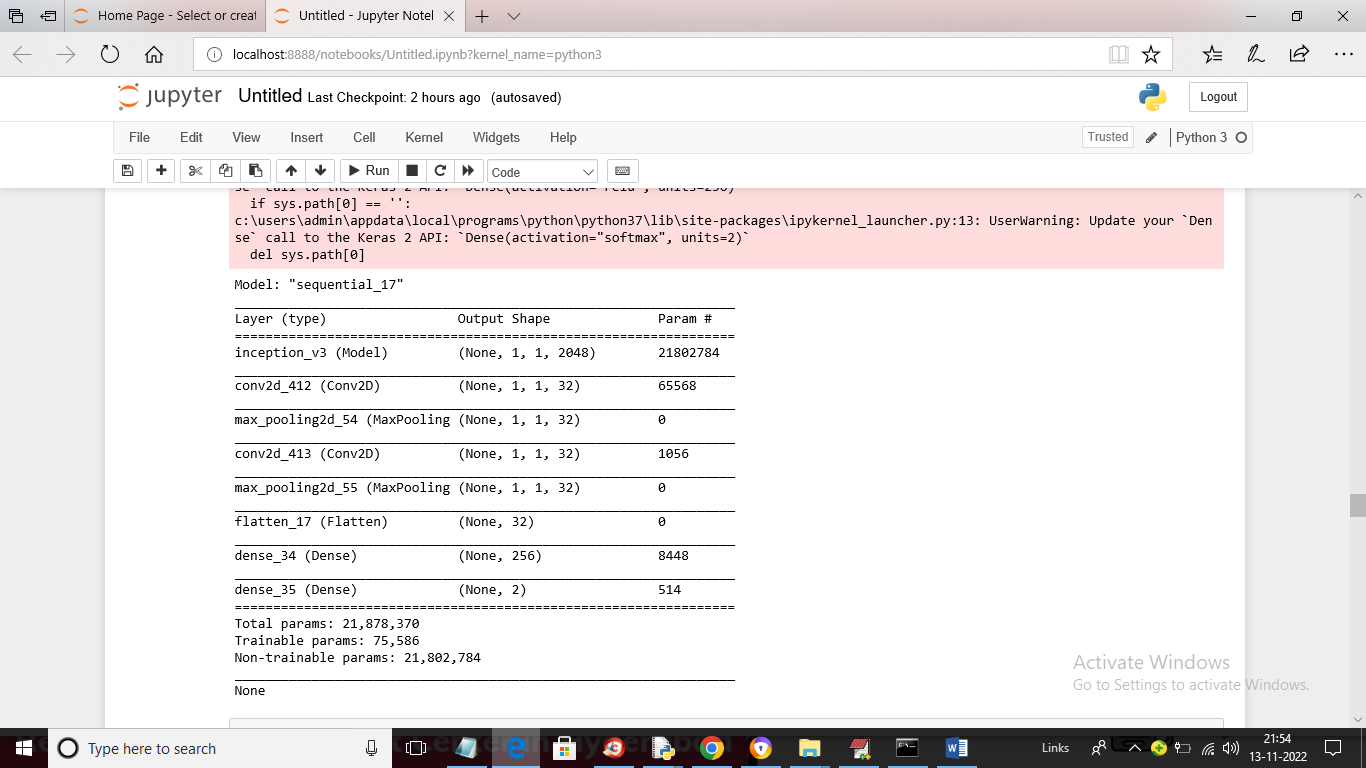
In above screen we can see Resnet101 layer details and below is the performance of Resnet101



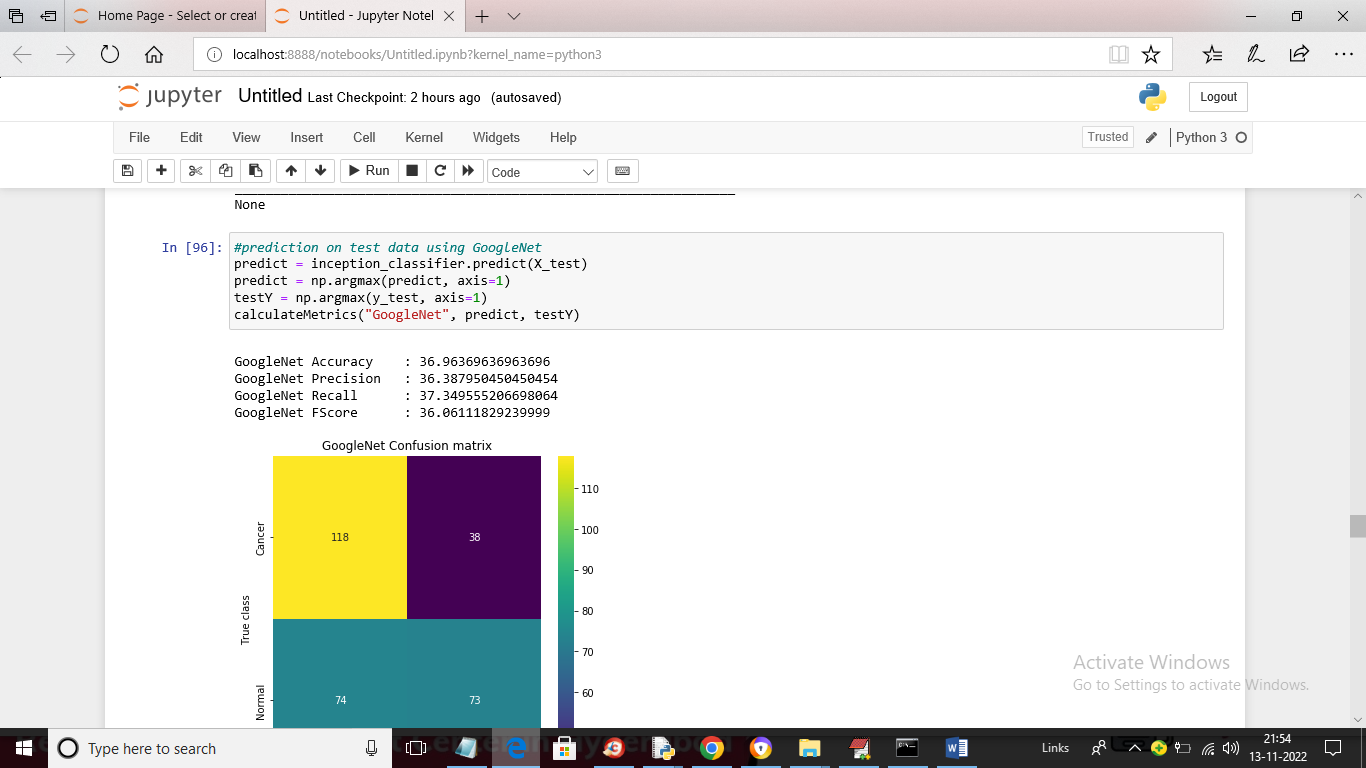
In above screen with Resnet101 we got 48% accuracy



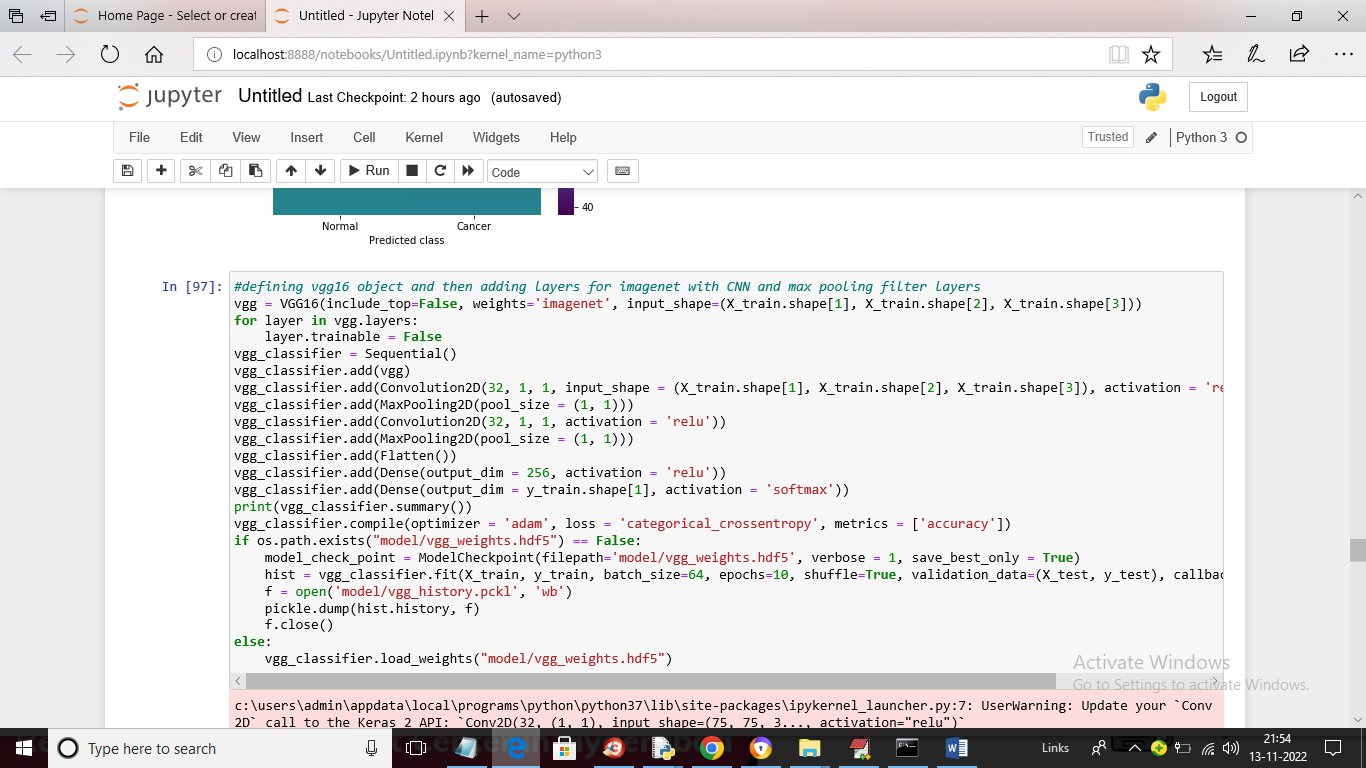
In above screen we are training with GoogleNet InceptionV3 model and below is the layer details



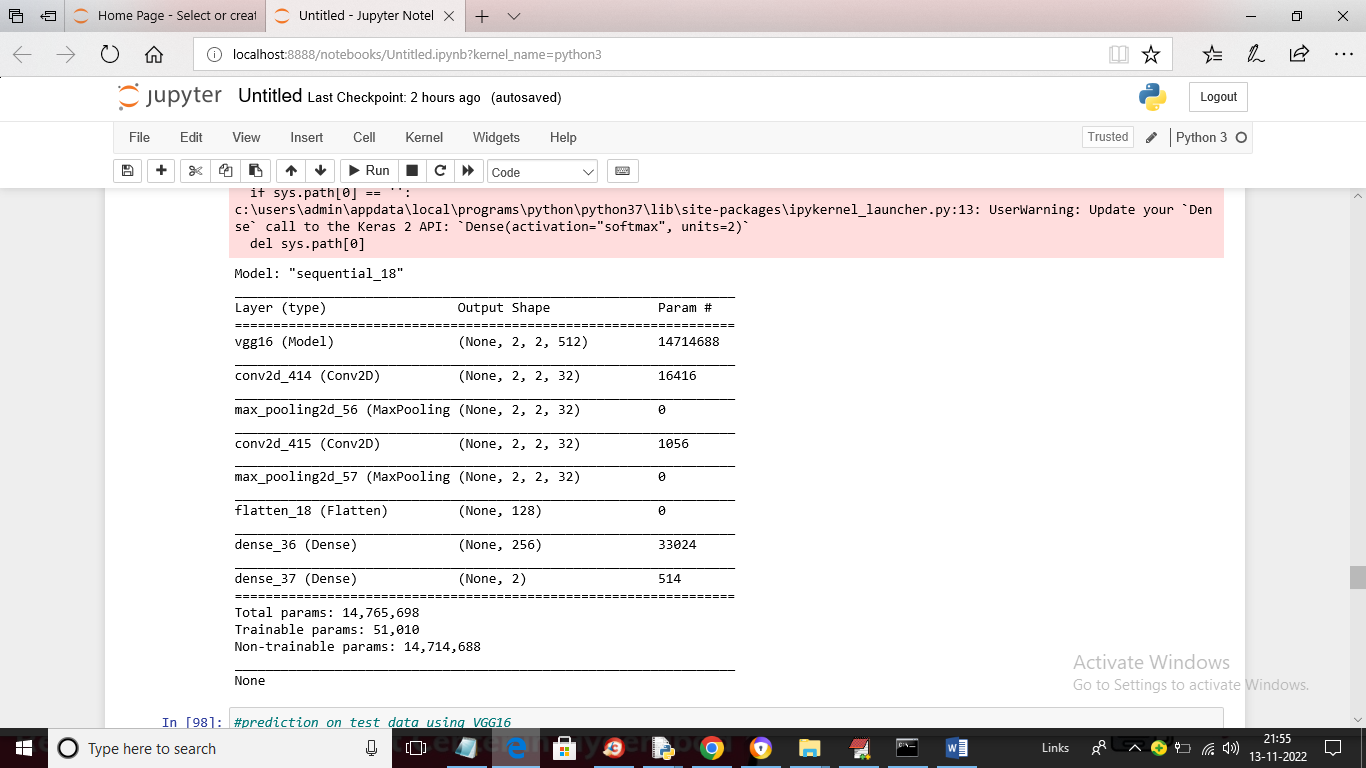
In above screen we can see GoogleNet layer details



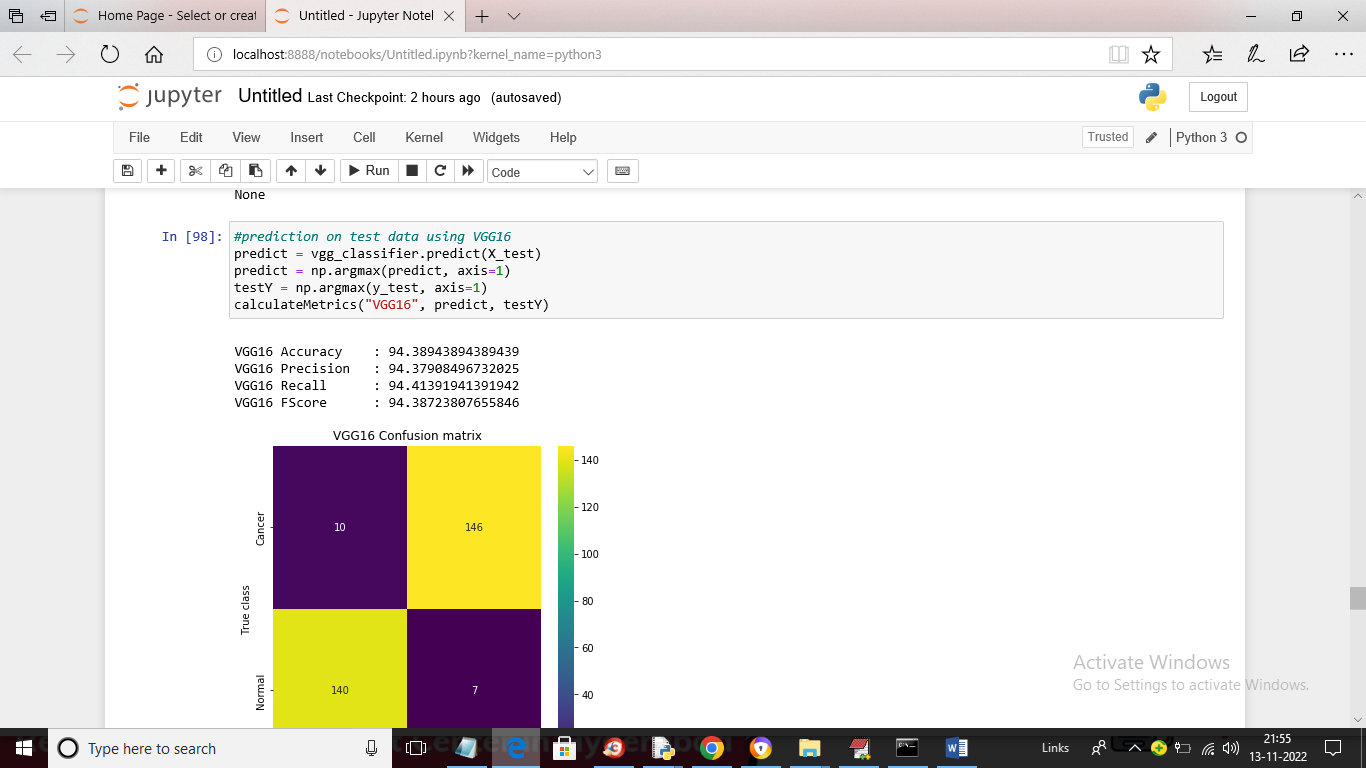
In above screen with GoogleNet we got 36% accuracy on same dataset



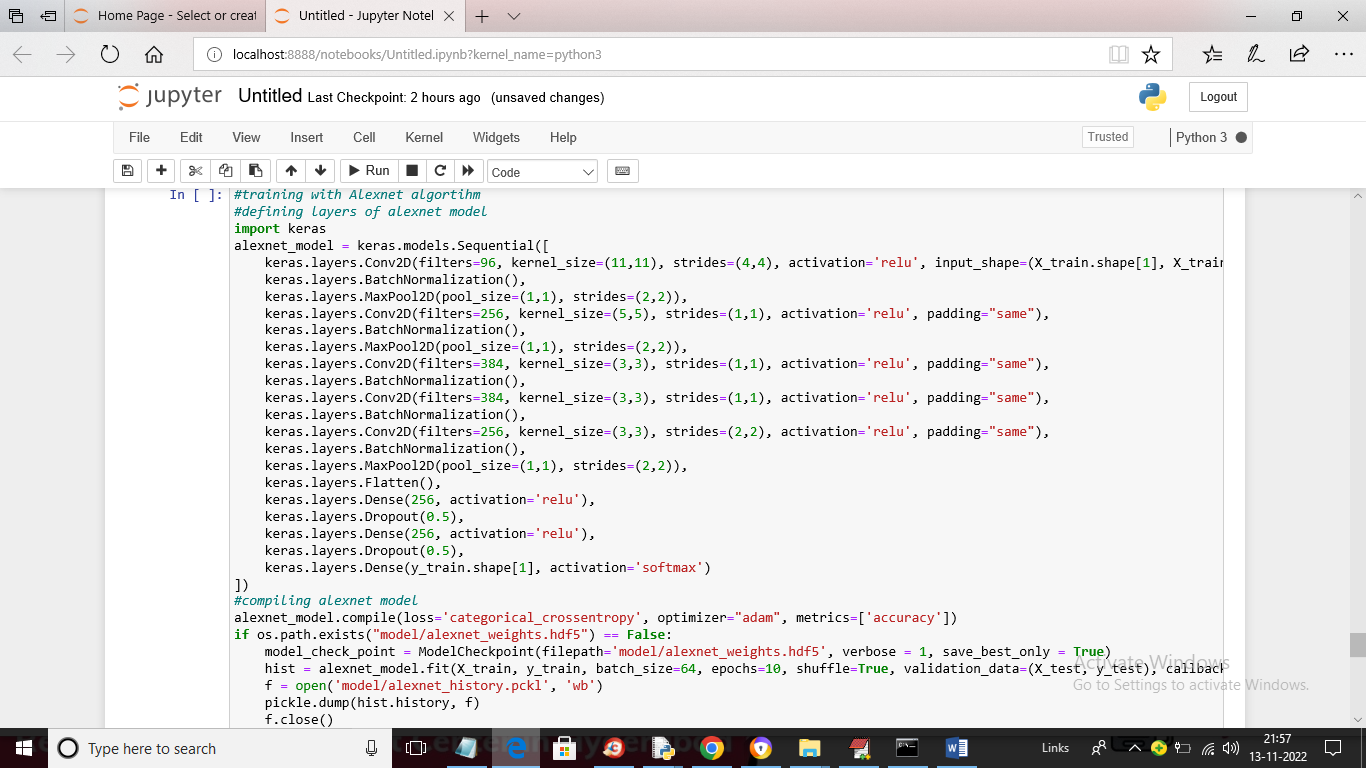
In above screen we are training with VGG16



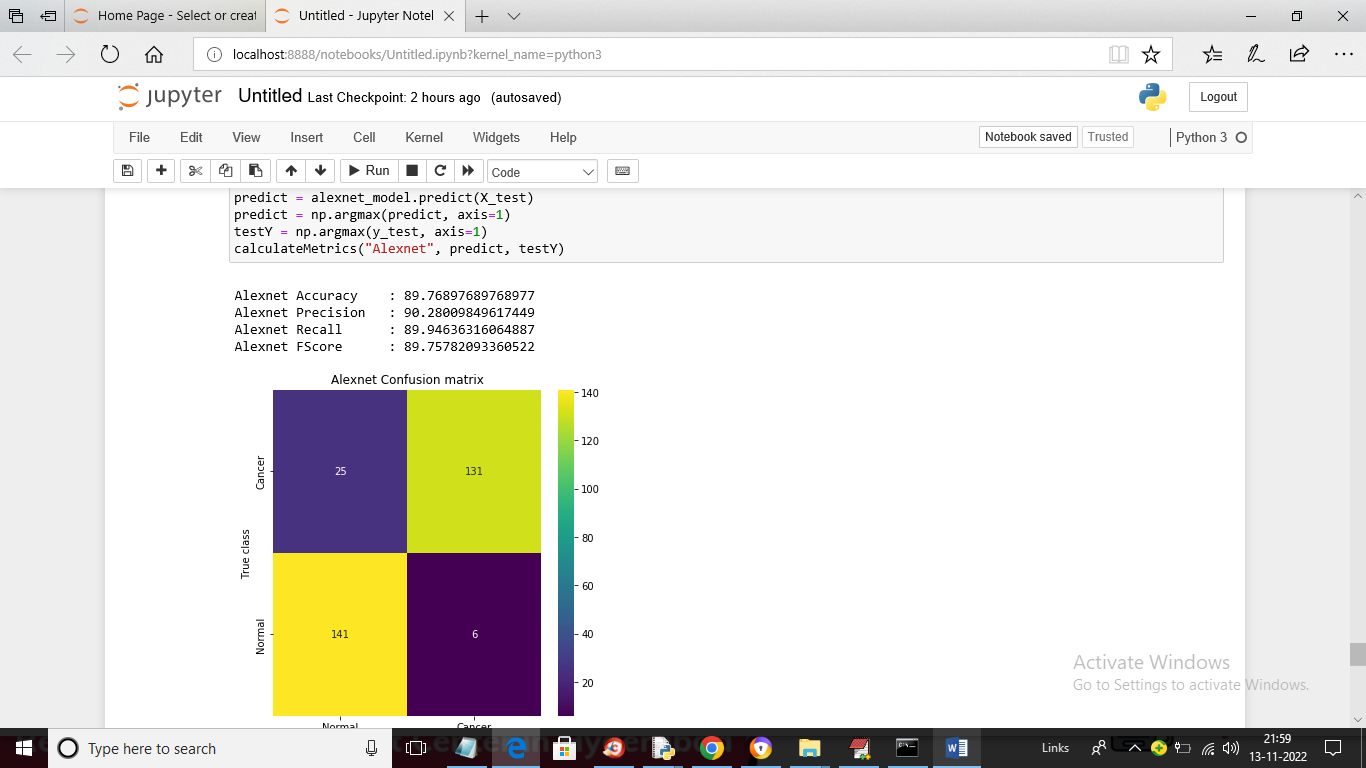
In above screen we can see VGG architecture layer details and below is the prediction performance



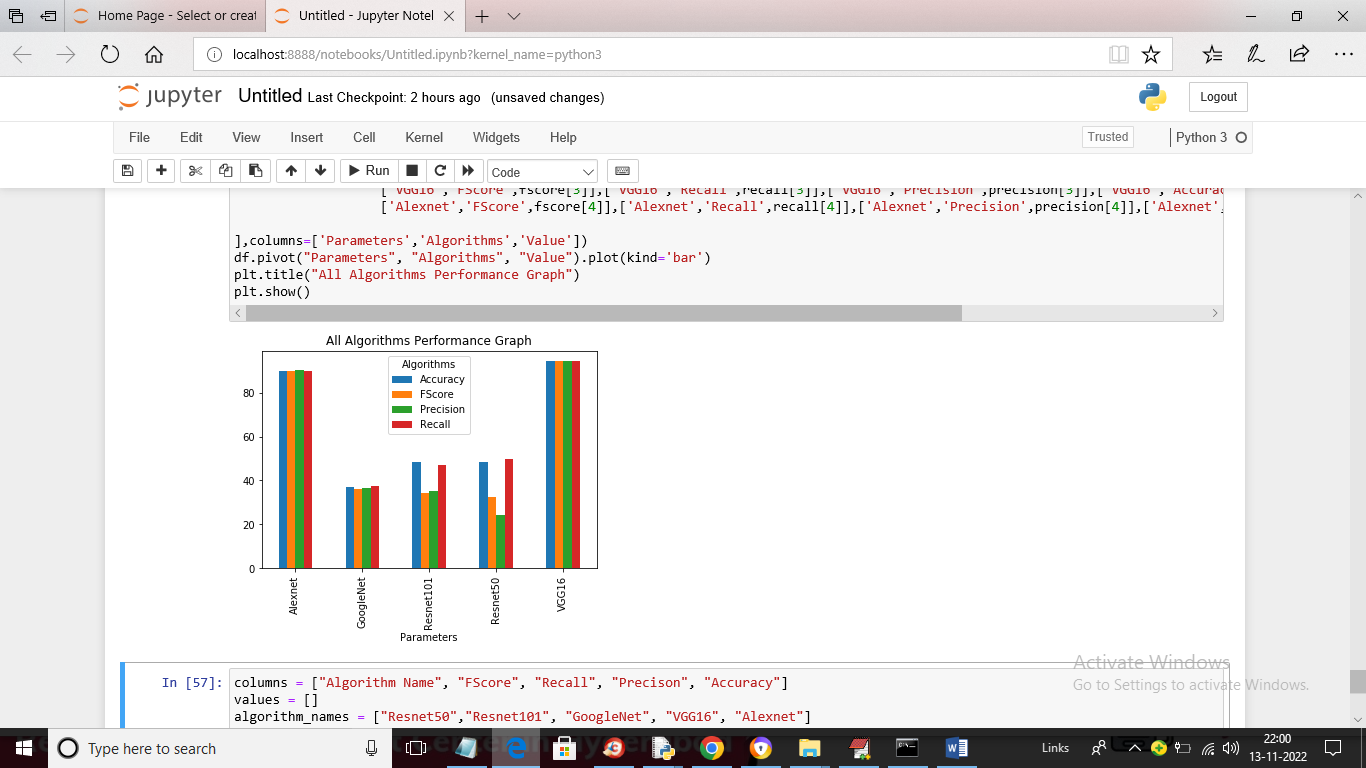
In above screen with VGG16 we got 94% accuracy and we can see blue colour boxes in confusion matrix contains incorrect prediction count and yellow colour boxes contains correct prediction count and in above graph we can see VGG16 predicted very few images incorrectly



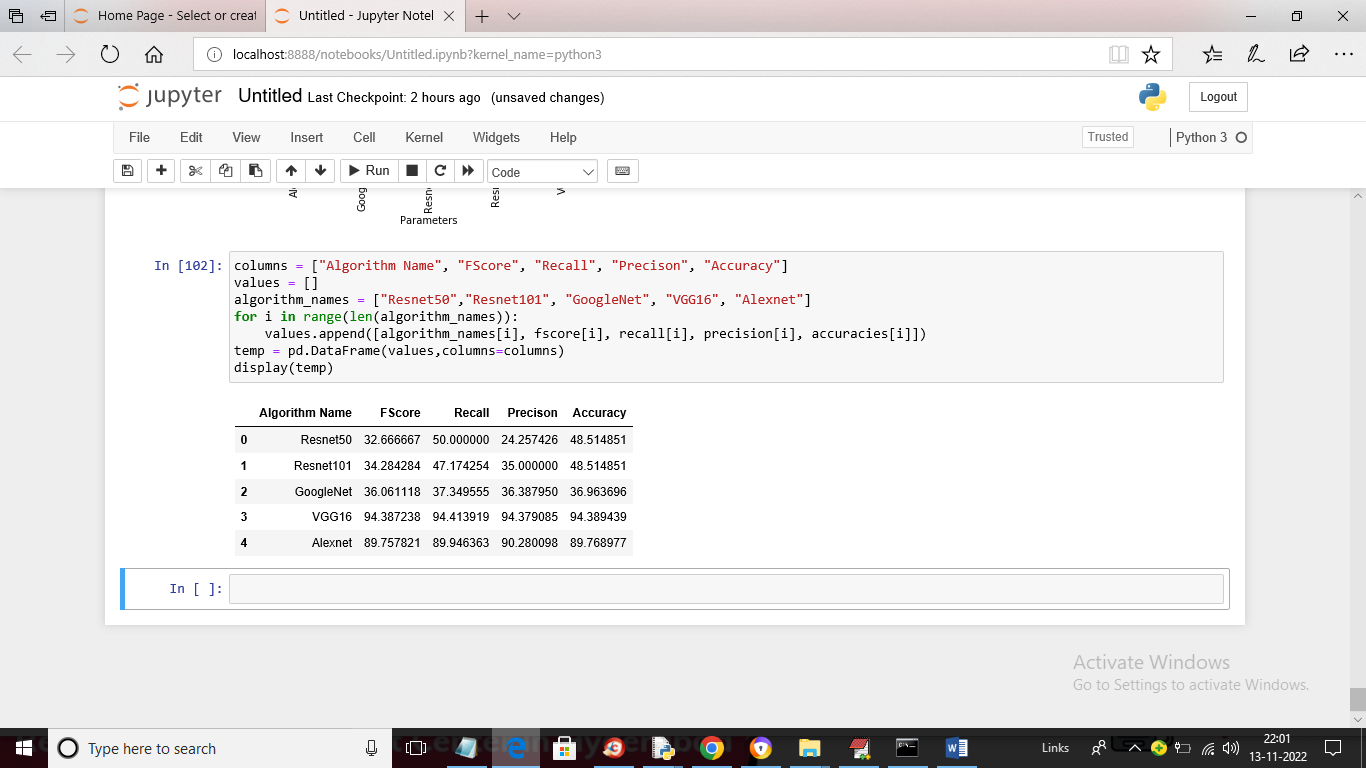
In above screen we are training with AlexNet model and below is the layer details



In above screen with AlexNet e got 89% accuracy and this accuracy may vary from 85 to 95% for different execution and below screen showing graphical based performance or each algorithm



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics values in different colour bar and in above graph we can see VGG and AlexNet got high performance and in below screen we can see above values in tabular format



In above screen we can see accuracy and other values in tabular format for each algortihms