Fish Disease Detection using CNN

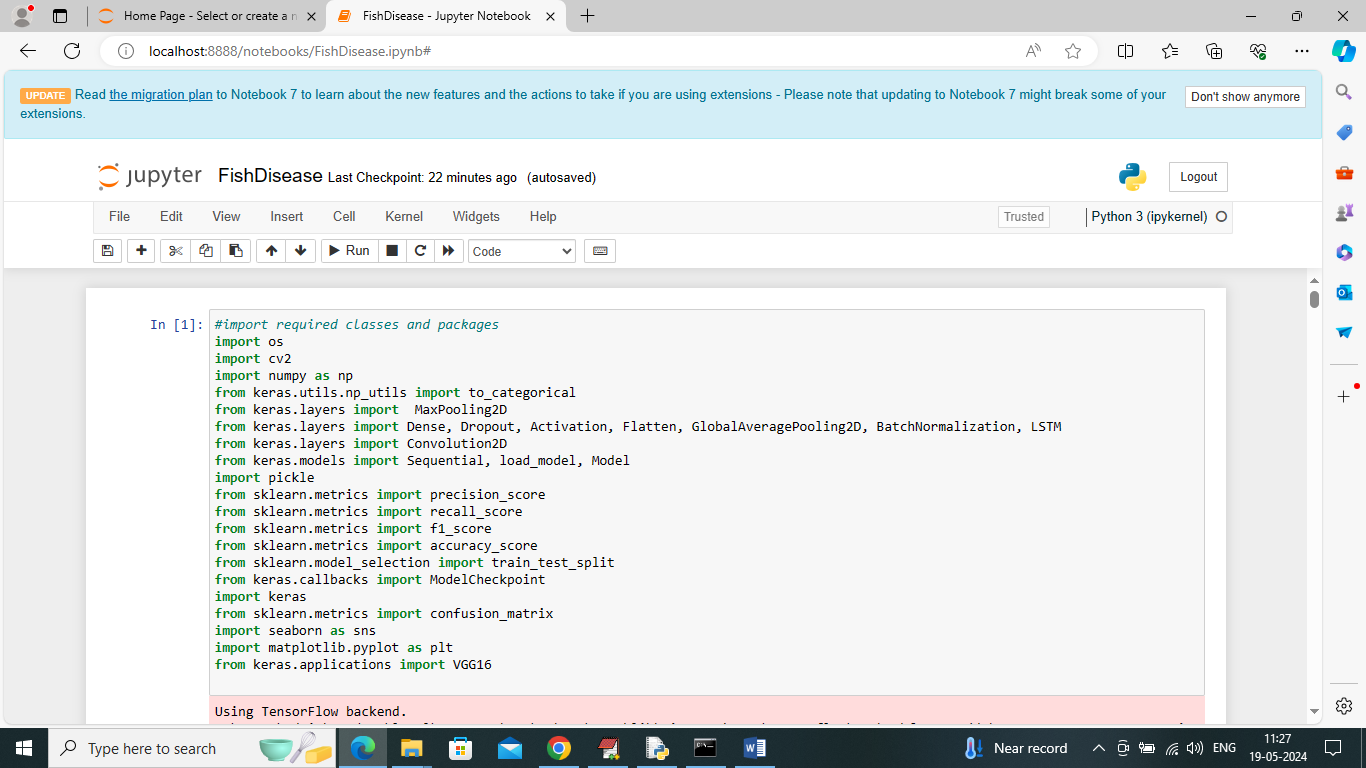
Fishes are the major revenue generator for any developed countries as this consume by maximum peoples across worlds. So government need to concentrate on fish’s health to avoid diseases and to maintain revenue for country development.

To identify fish disease type we are employing AI algorithms such as CNN, pre-trained VGG16 and LSTM algorithms. This algorithms gain considerable amount of popularity in almost all fields such as health care, banking and many more. This algorithms can get trained on possible fish disease images and this trained model can be applied on fish image to detect weather fish contains any disease or healthy.

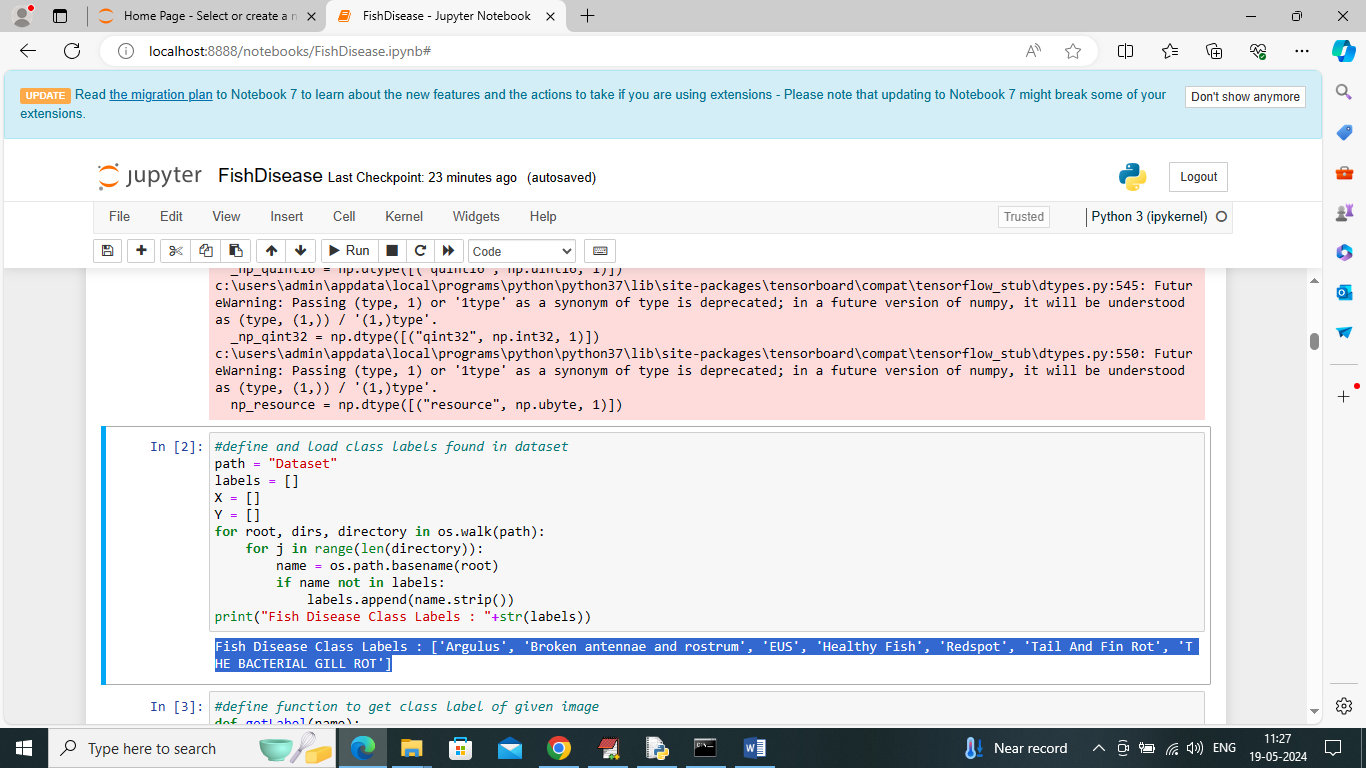
To train above algorithms we have utilized fish disease dataset from KAGGLE repository <https://www.kaggle.com/datasets/utpoldas/freshwater-fish-disease-dataset>. This dataset contains 6 different types of diseases excluding healthy images.

All algorithms get trained on above dataset and each algorithm performance is evaluated in terms of accuracy, precision, recall, FSCORE and confusion matrix. Among all algorithms CNN is giving best accuracy.

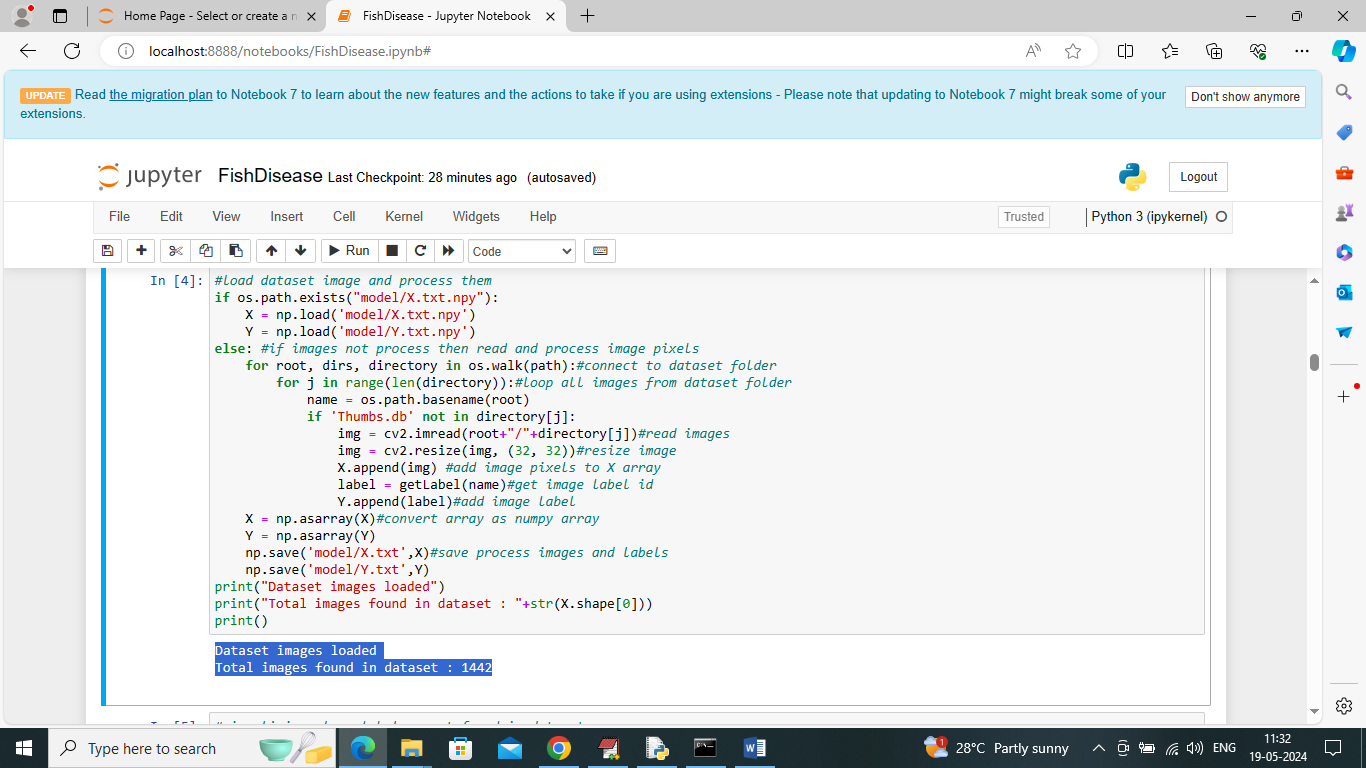
We have coded this project using JUPYTER notebook and below are the code and output screens with blue color comments



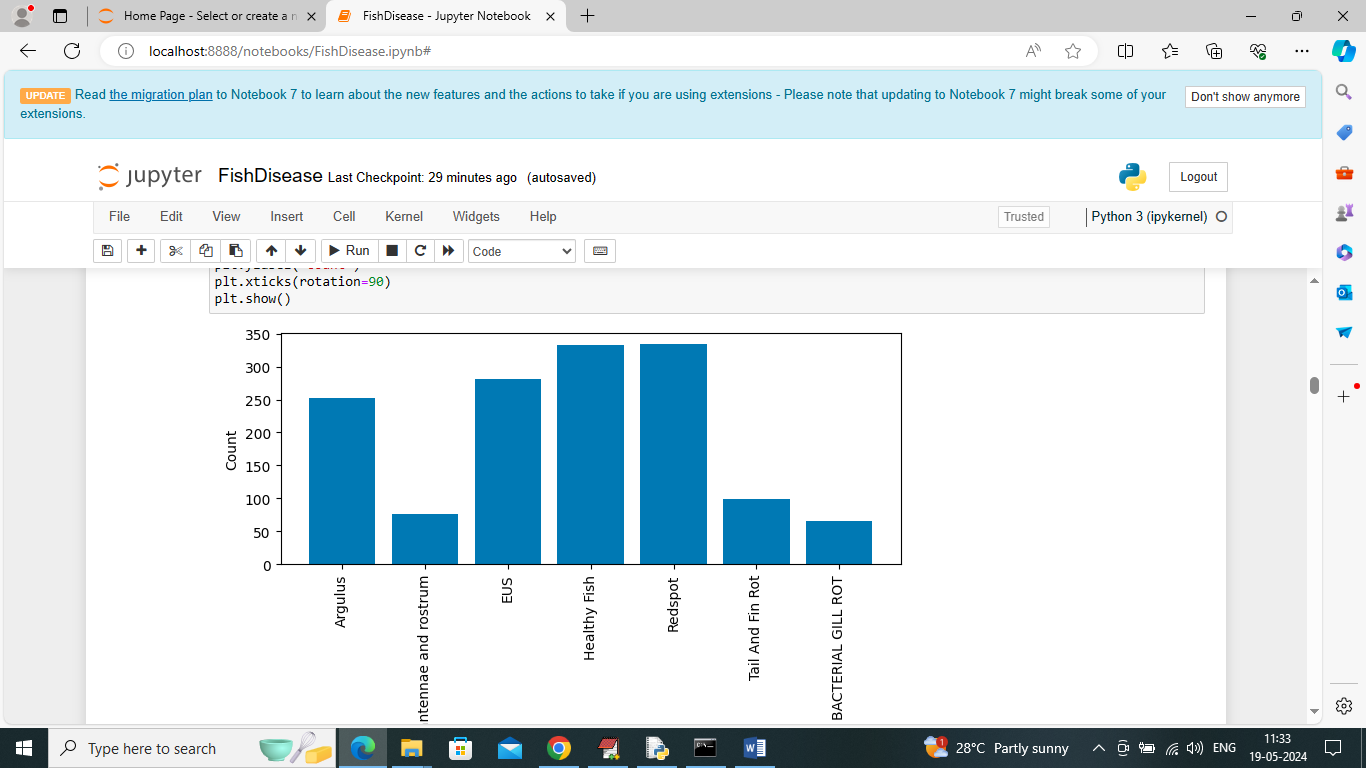
In above screen importing required python classes and packages



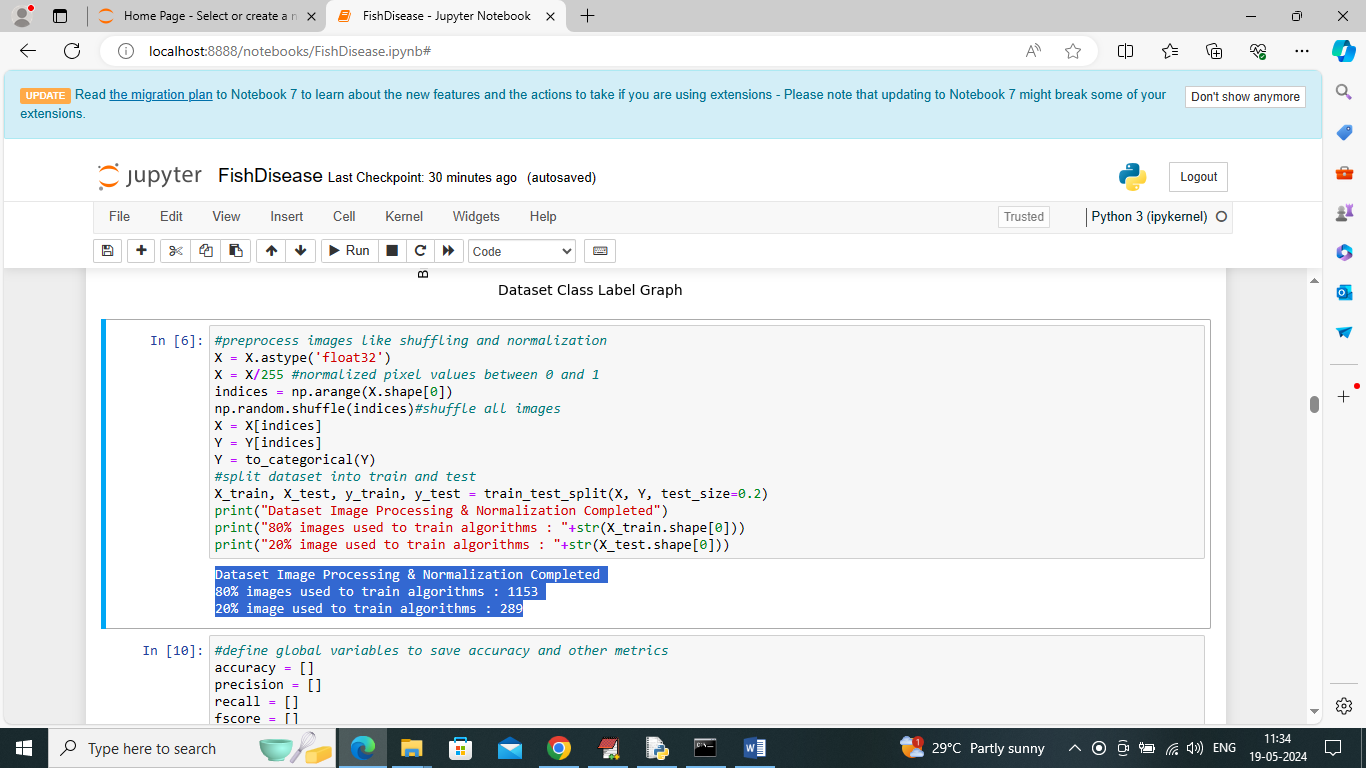
In above screen finding and displaying different fish disease names found in dataset and this names you can see in blue color text



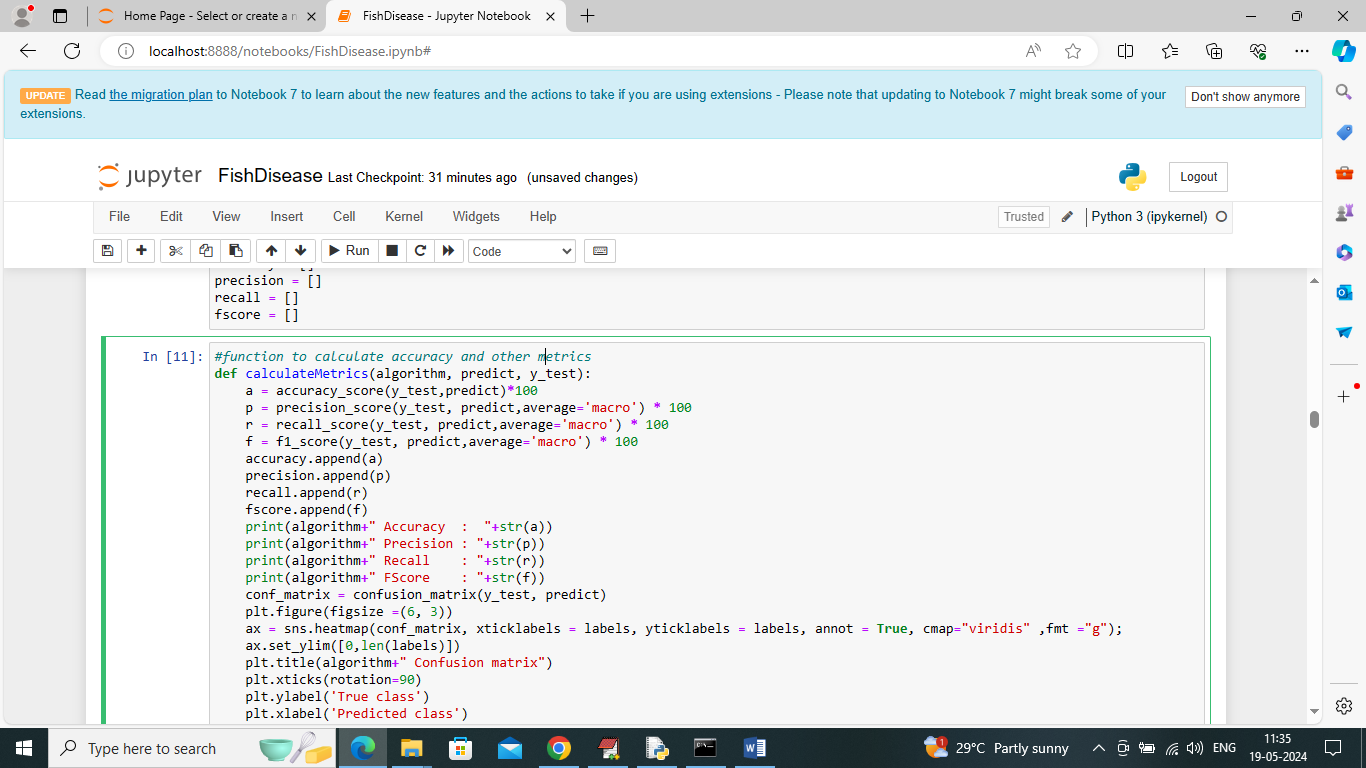
In above screen reading all images from dataset and then resizing them to equal sizes and then creating X and Y training array where X will contains image features and Y will contains fish disease label and then displaying total number of loaded images



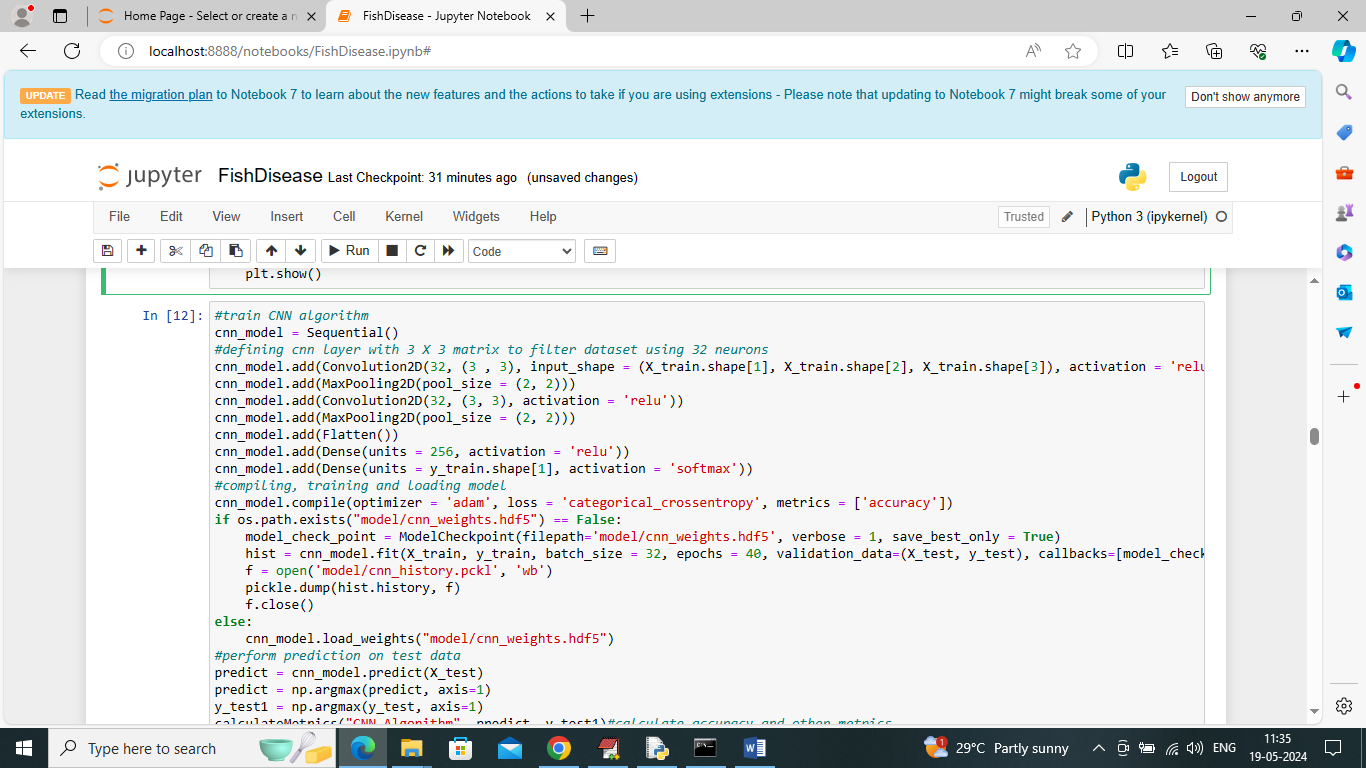
In above graph x-axis represents fish disease names and y-axis contains count of fish disease images found in that category



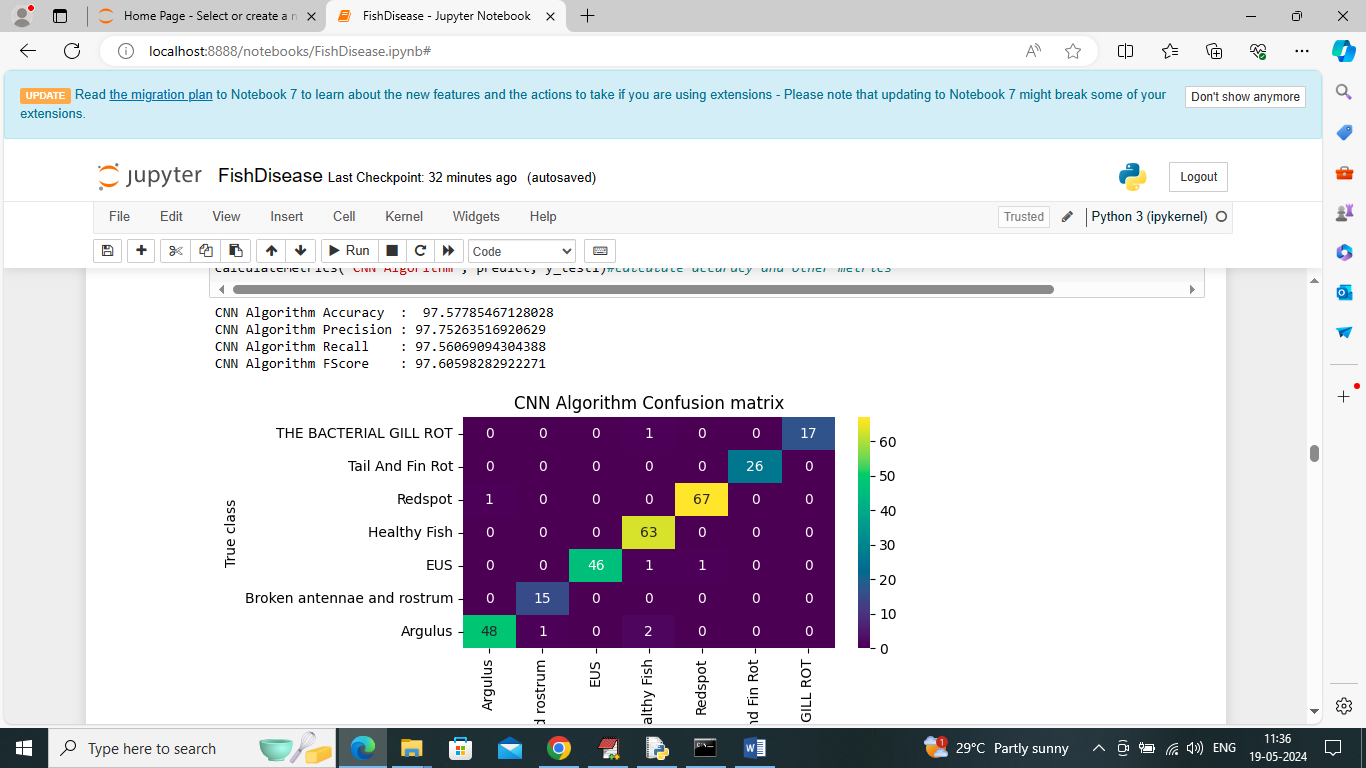
In above screen applying image processing techniques such as shuffling, normalization and then splitting all images in to train and test where application using 80% images for training and 20% for testing



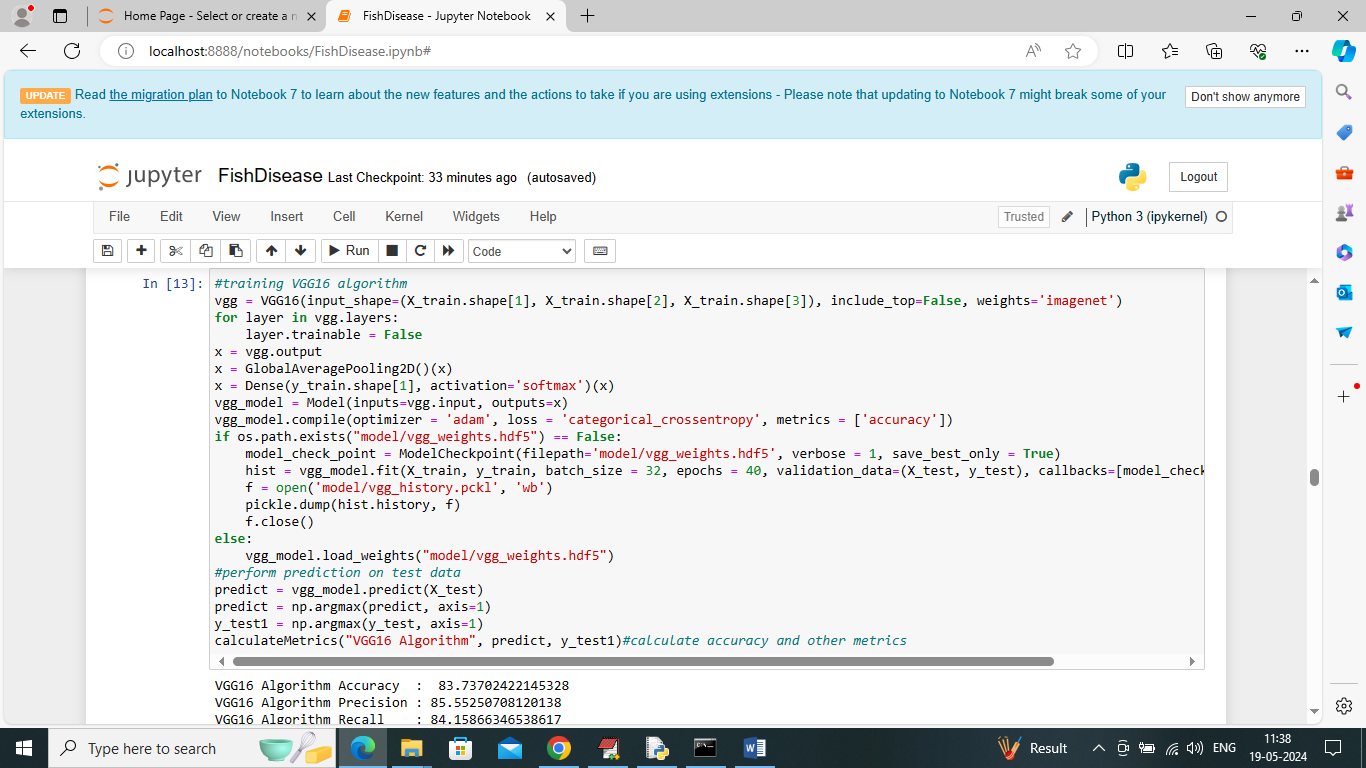
In above screen defining function to calculate accuracy and other metrics



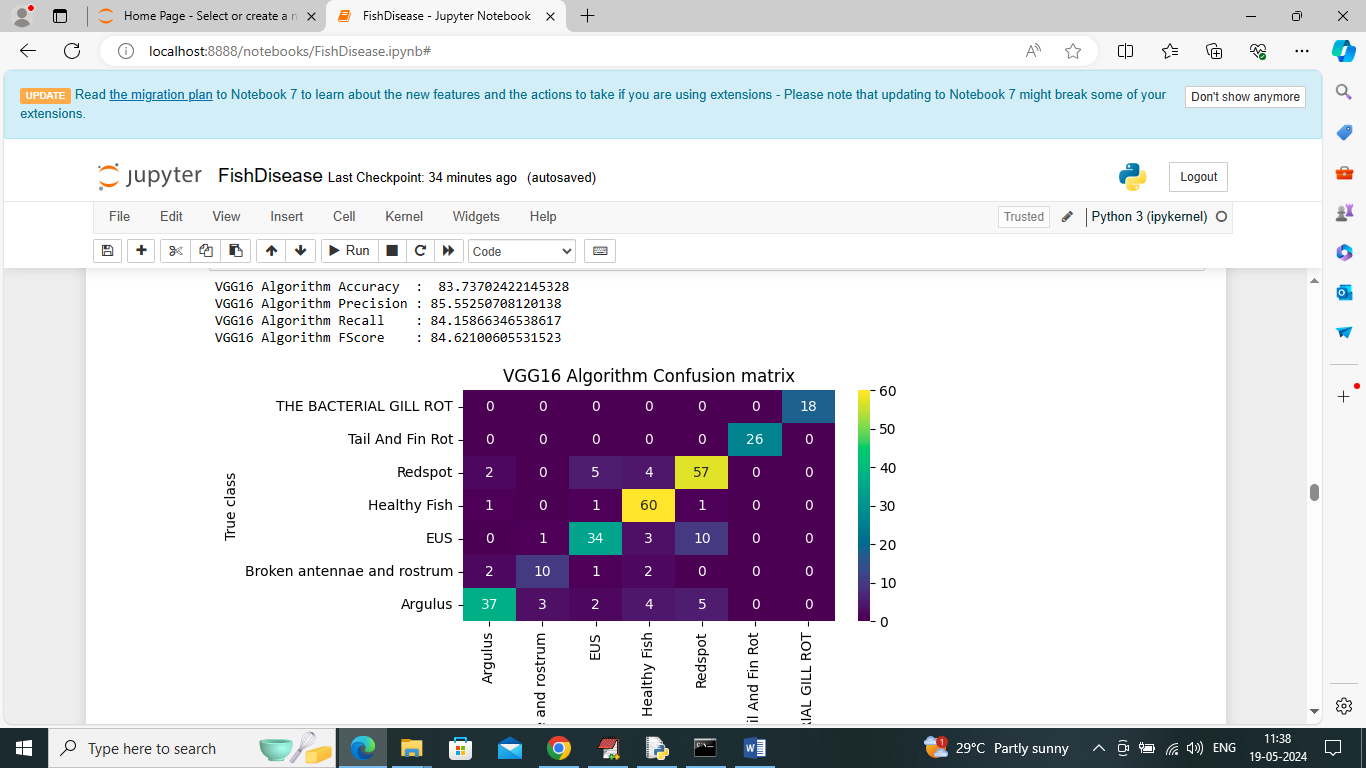
In above screen defining layers of CNN algorithm and after training and executing this block will get below output



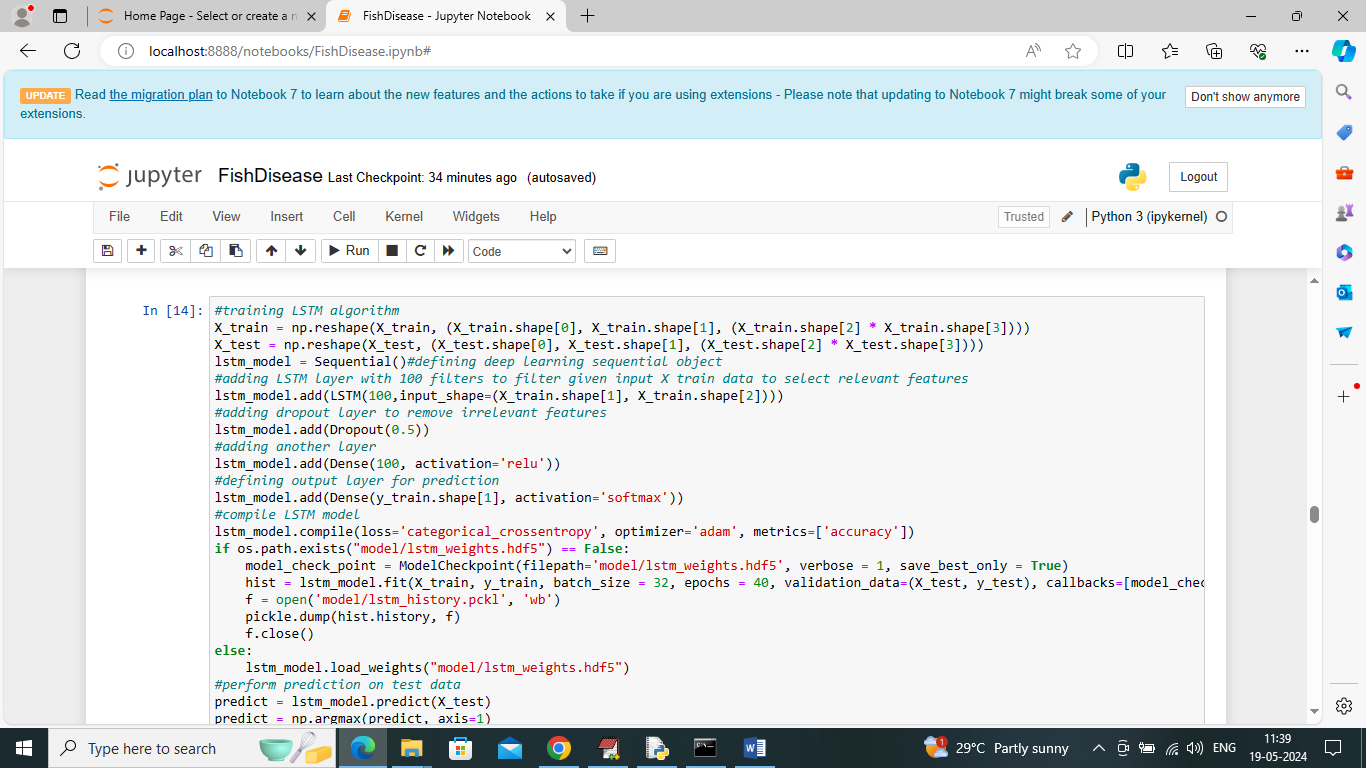
In above screen CNN algorithm got 97% accuracy and can see other metrics like precision, recall and FSCORE. In confusion matrix graph x-axis represents ‘Predicted Class Labels’ and y-axis represents true class labels and then all different color boxes in diagnol represents correct prediction count and remaining blue boxes contains incorrect prediction count which are very few.



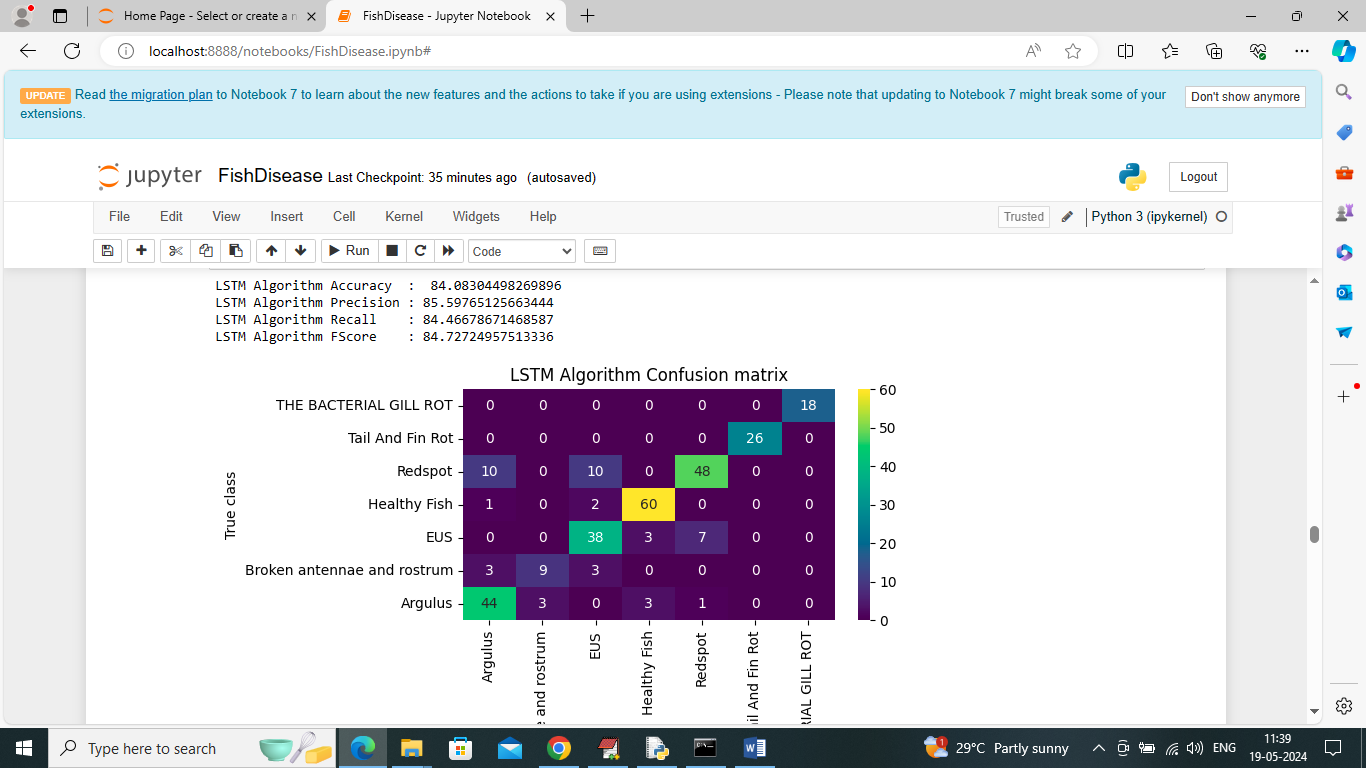
In above screen defining and training VGG16 algorithm and after executing above block will get below output



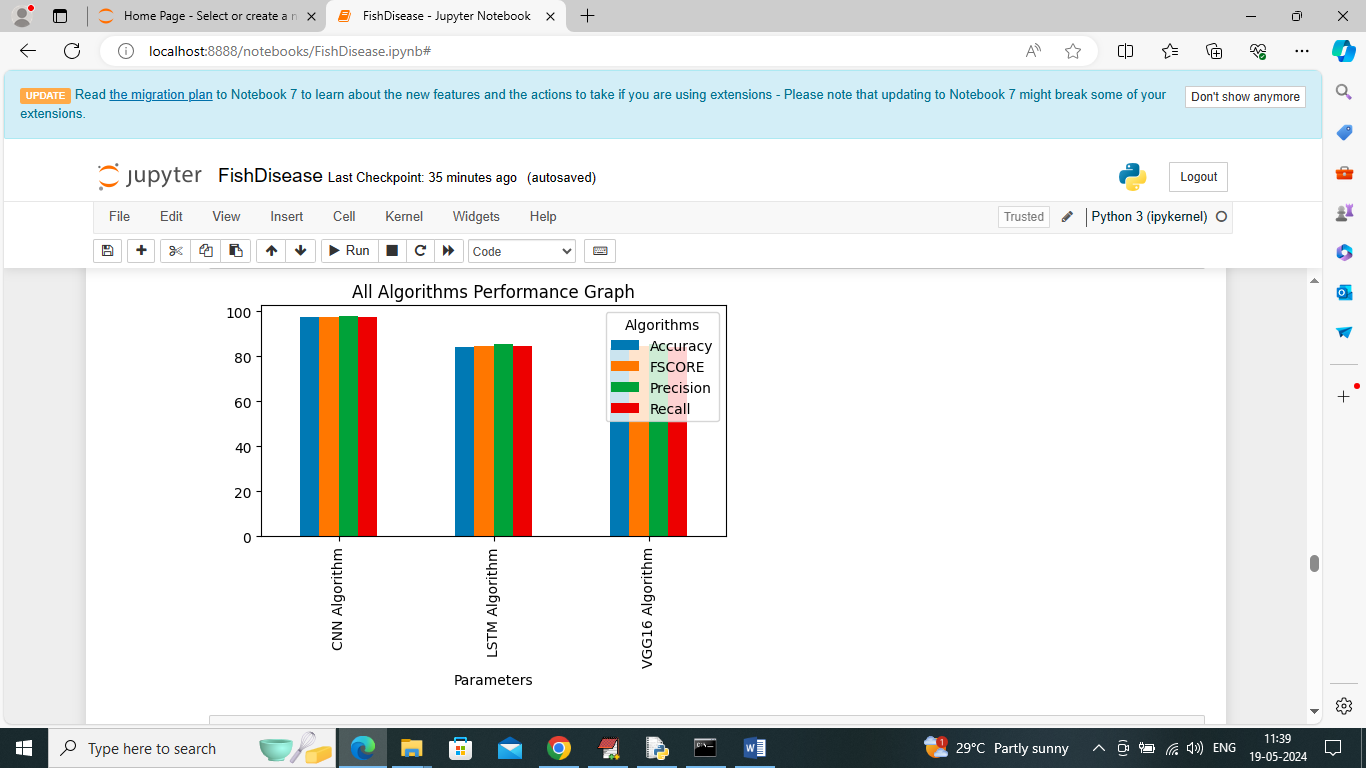
In above screen VGG algorithm got 83% accuracy and can see other metrics output also



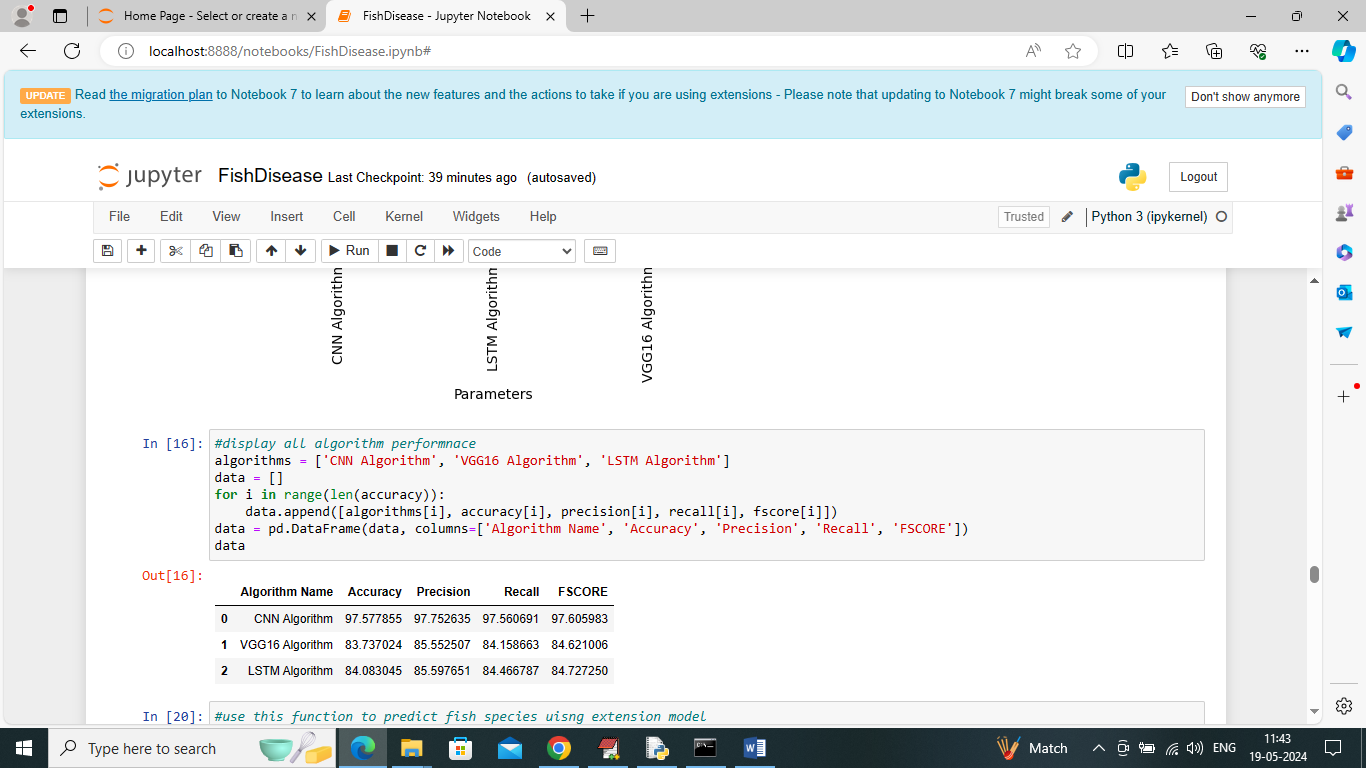
In above screen defining and training LSTM algorithm and after executing this block will get below output



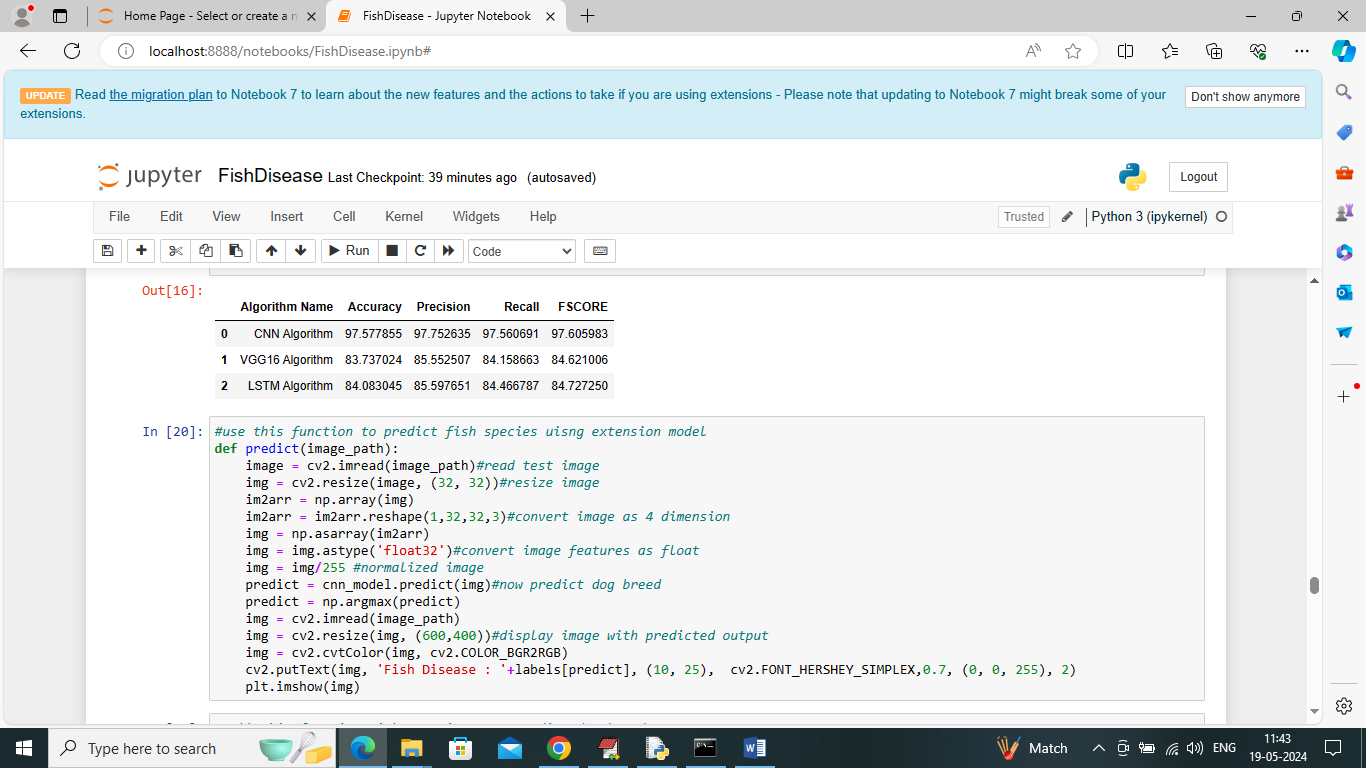
In above screen LSTM got 84% accuracy and can see other metrics output also



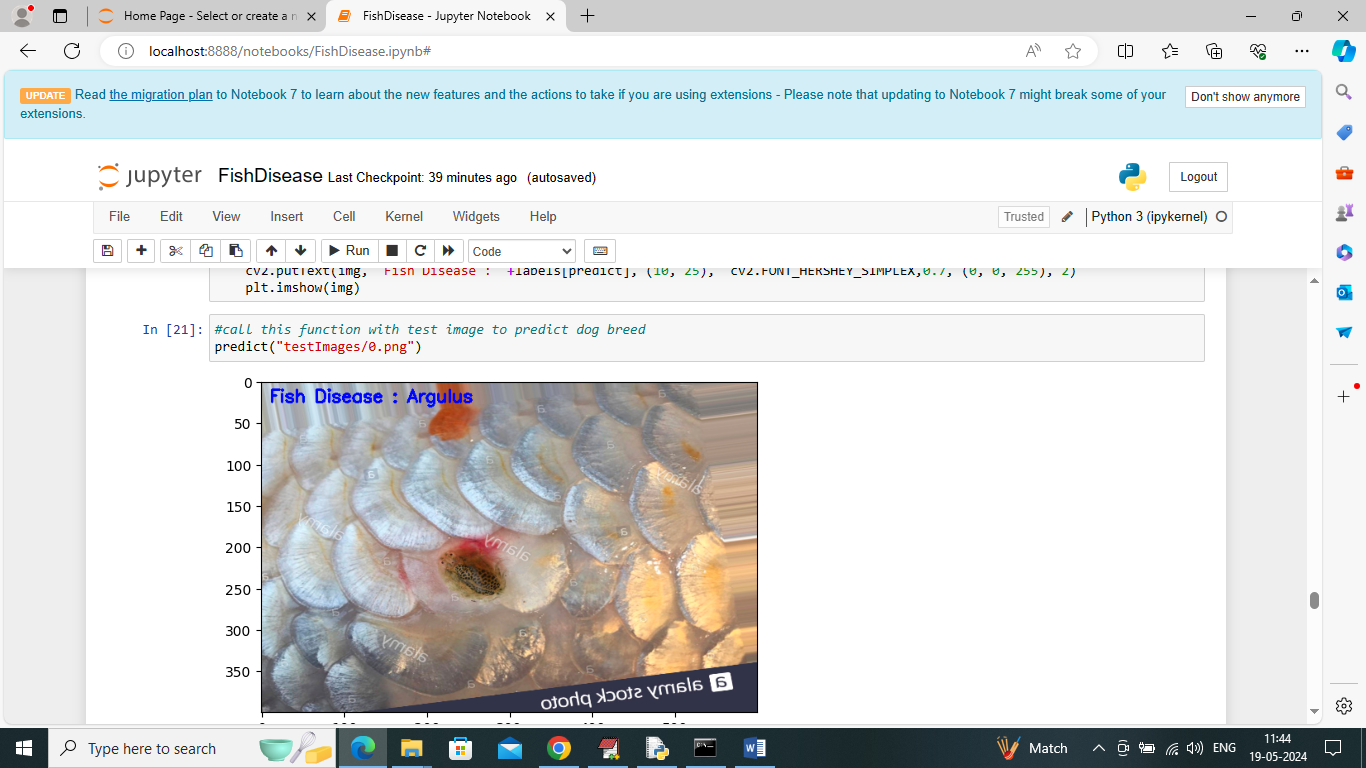
In above graph displaying comparison graph between all algorithms where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different color bars and in all algorithms CNN got high performance



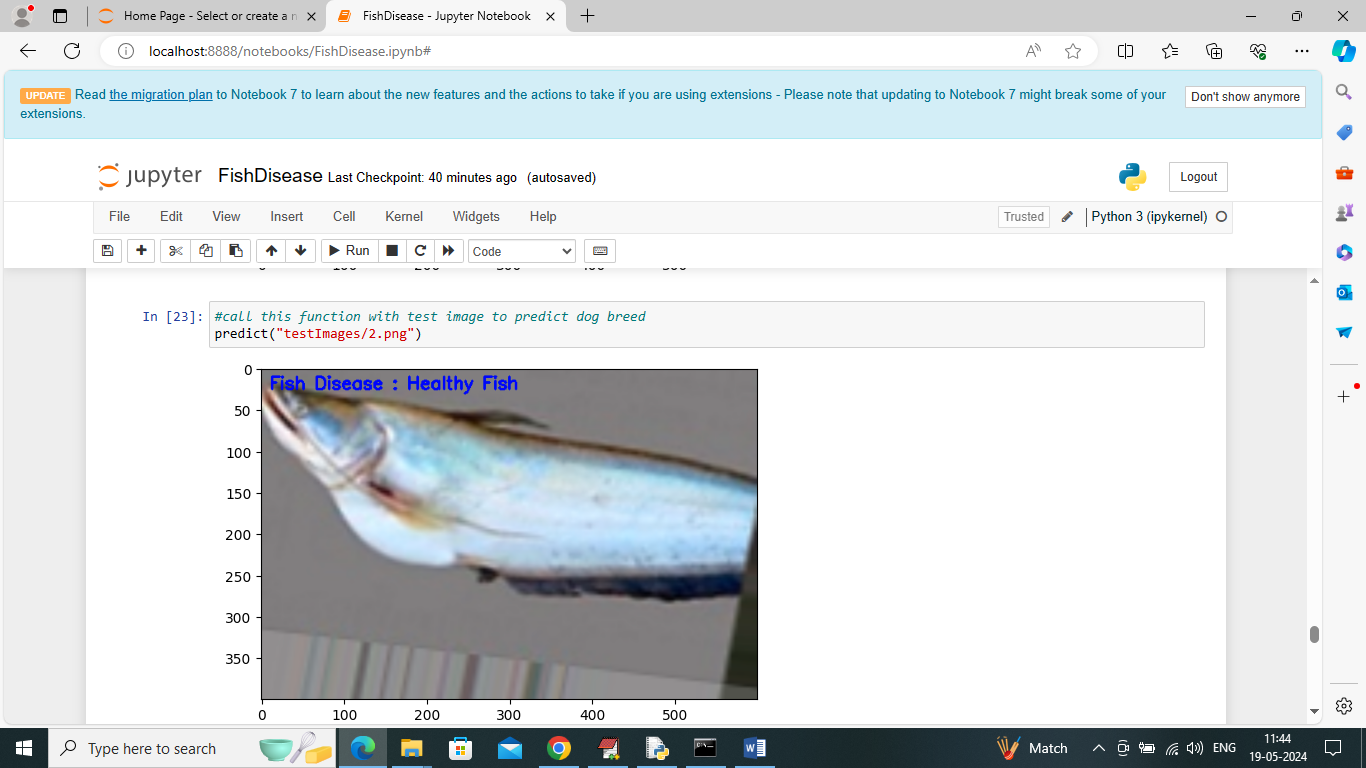
In above screen displaying all algorithm performance in tabular format



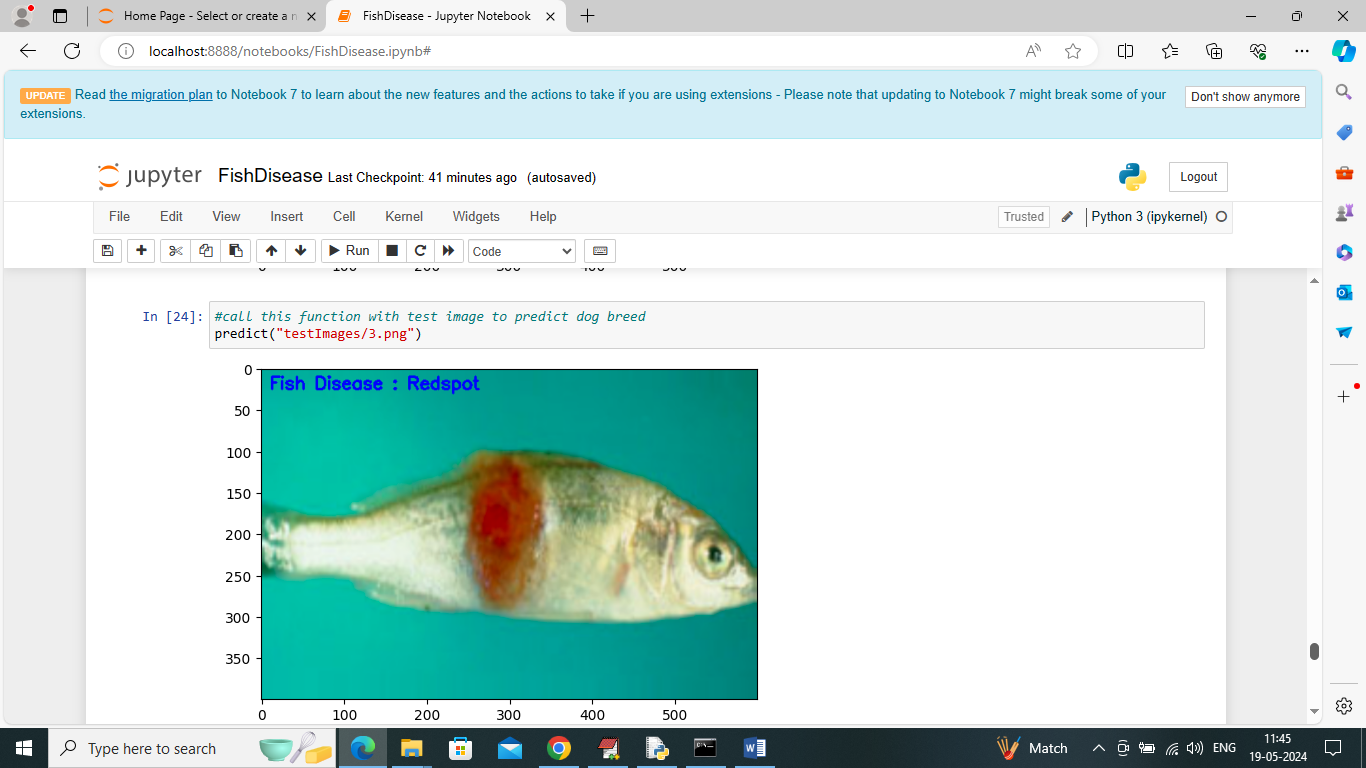
In above screen defining predict function which will read input image and then predict fish disease type



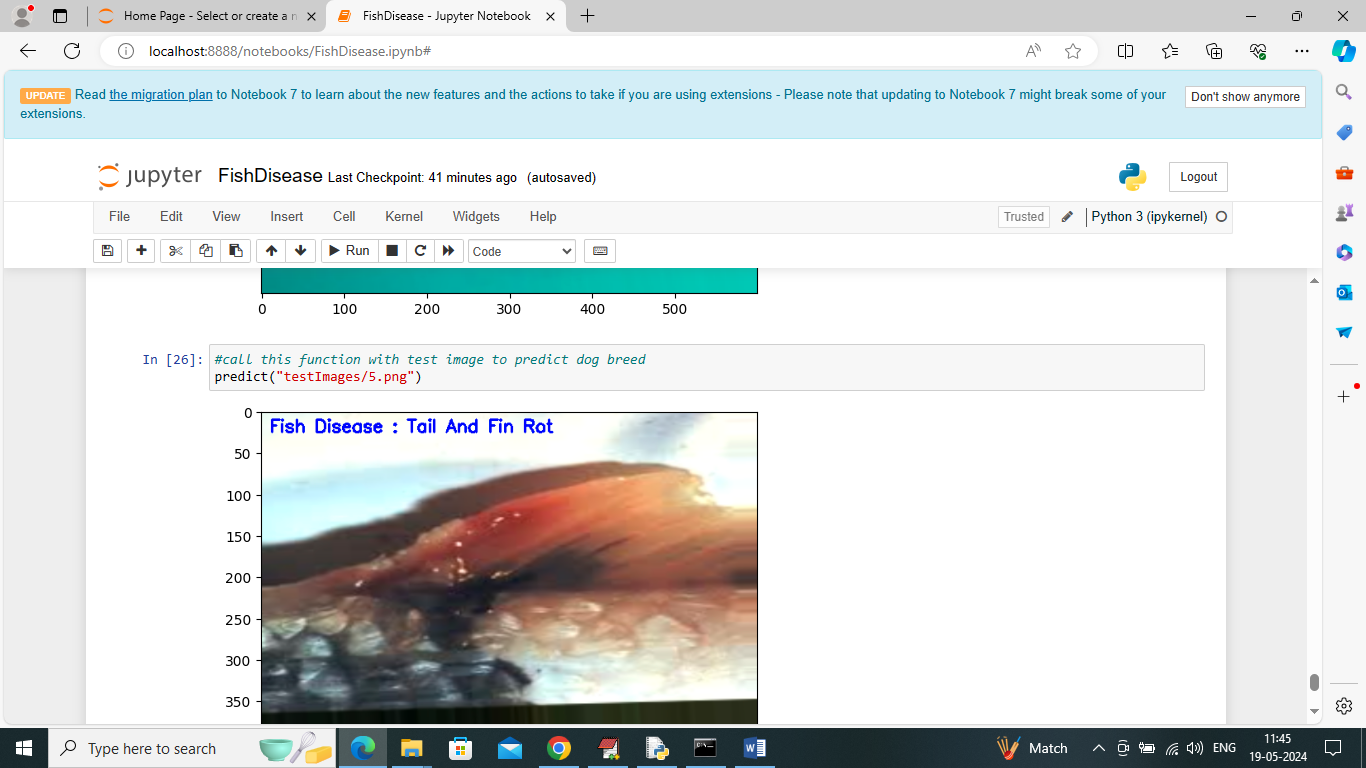
In above screen calling predict function with image path and then in output can see fish disease found in image predicted as ‘Argulus’



In above screen fish detected as ‘Healthy’



In above image disease detected as ‘Red spot’



In above screen disease detected as ‘Tail and Fin Rot’.

Similarly by giving image path we can predict fish disease type