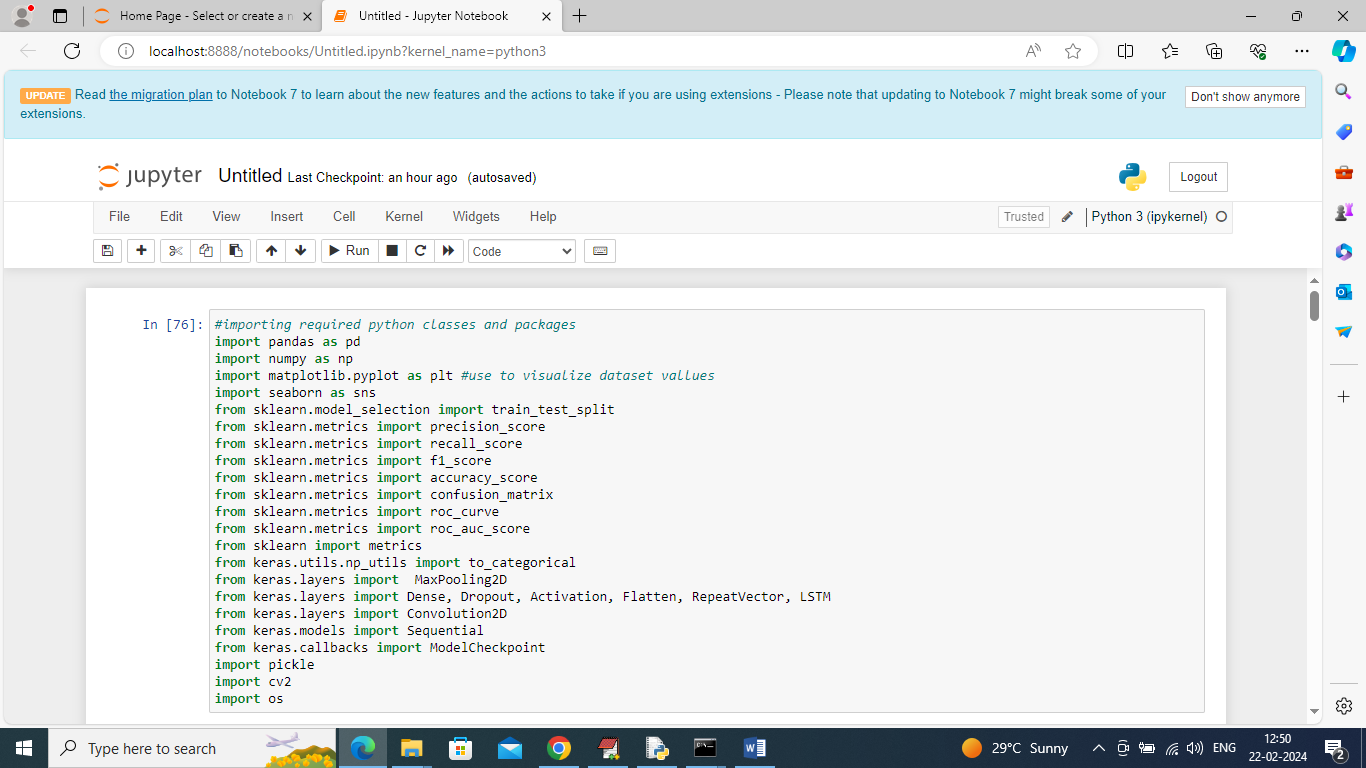
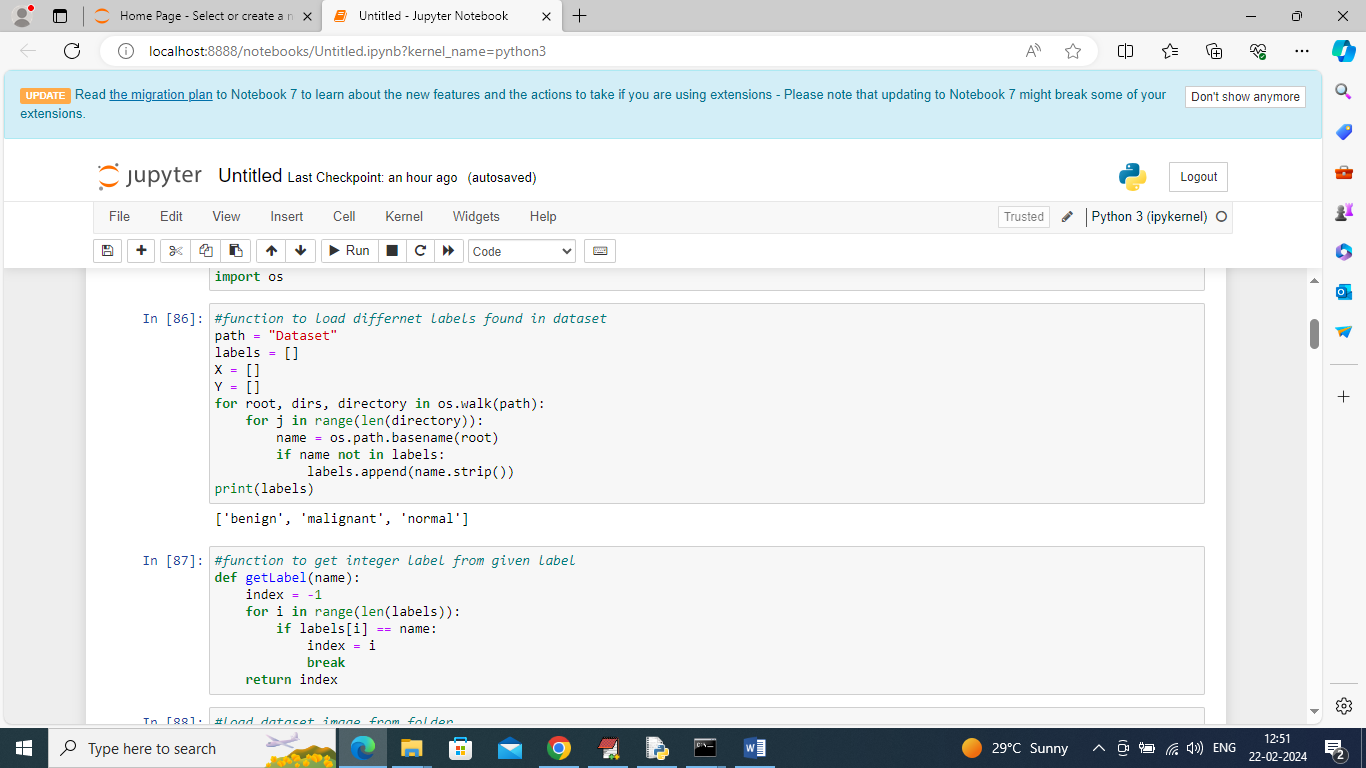
Breast cancer detection using CNN and RNN (hybrid)

As per your instruction we have developed combination of CNN & LSTM (RNN) to detect breast cancer disease and as an extra layer we have added RELU with Softmax. ReLU helps the first hidden layer receive errors from the last layers to adjust all weights between layers and Softmax layer which will divide each class prediction into probabilities and the class with highest probability will be best prediction and help in enhancing accuracy.

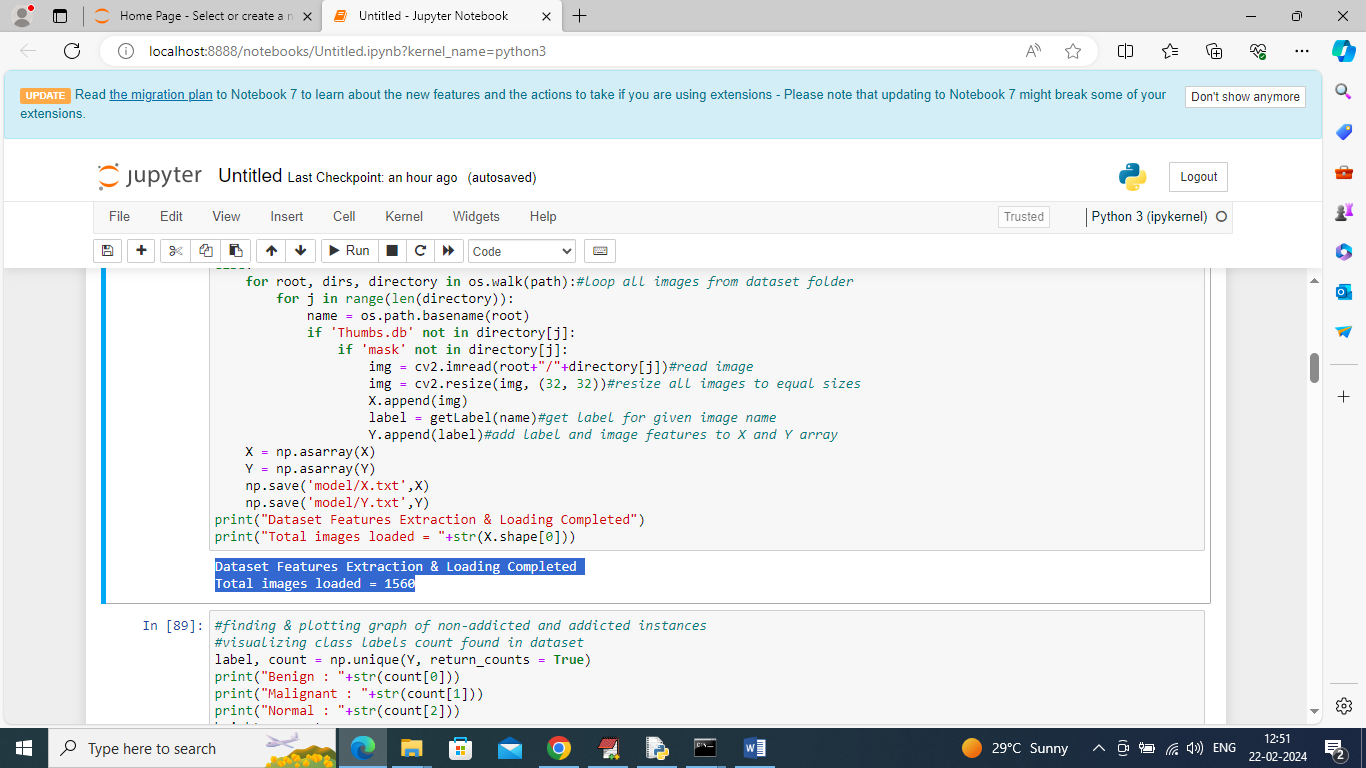
To train above algorithms we have utilized same dataset given by your and we have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments. All processing and model training we done using JUPYTER and disease prediction will be running from WEB.



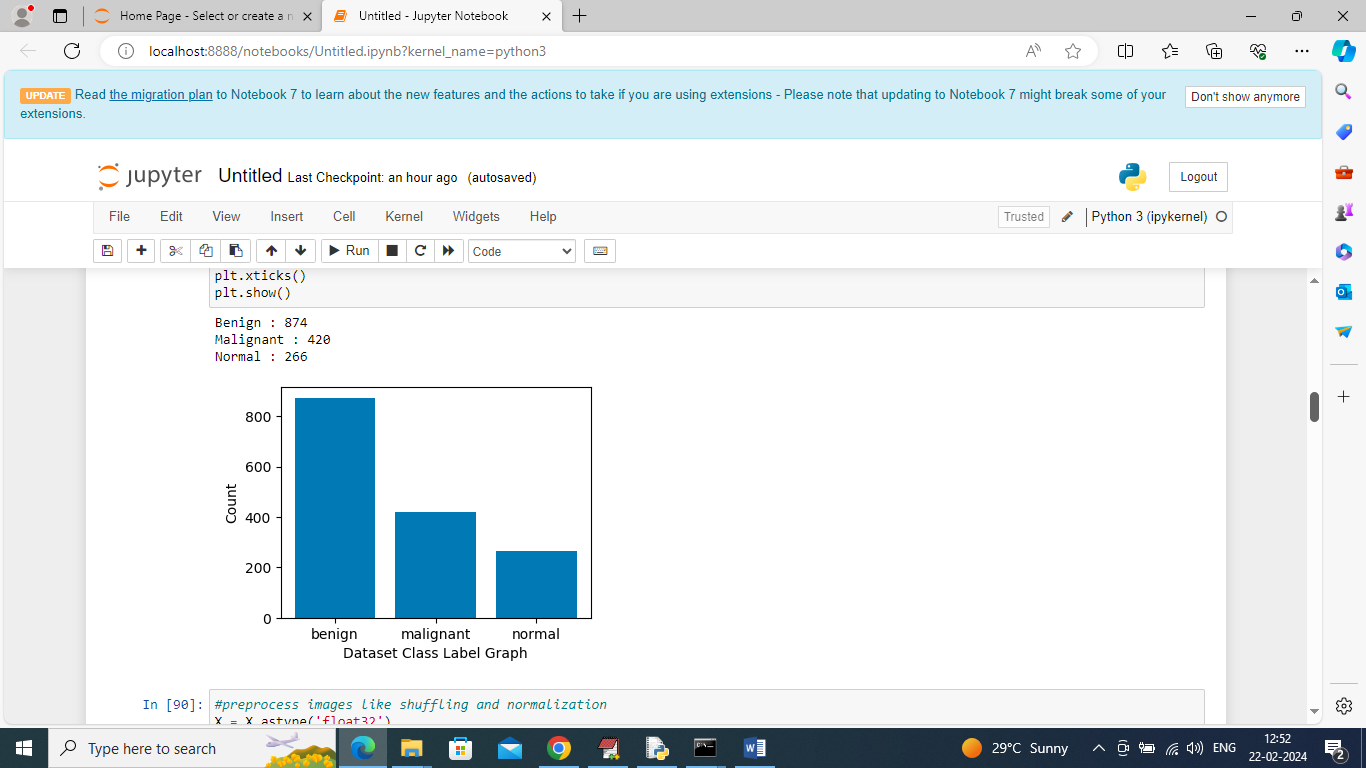
In above screen importing required python packages and classes



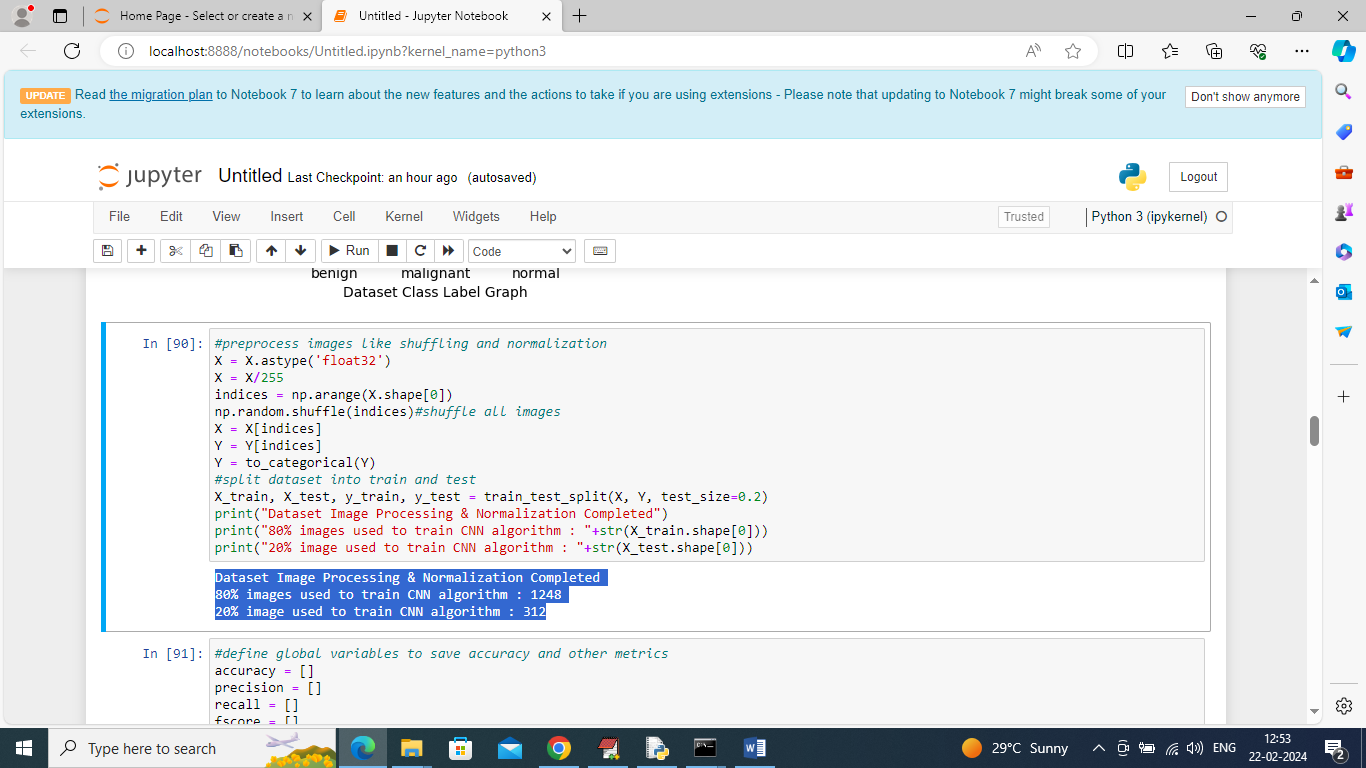
In above screen defining function to find and display different labels found in dataset



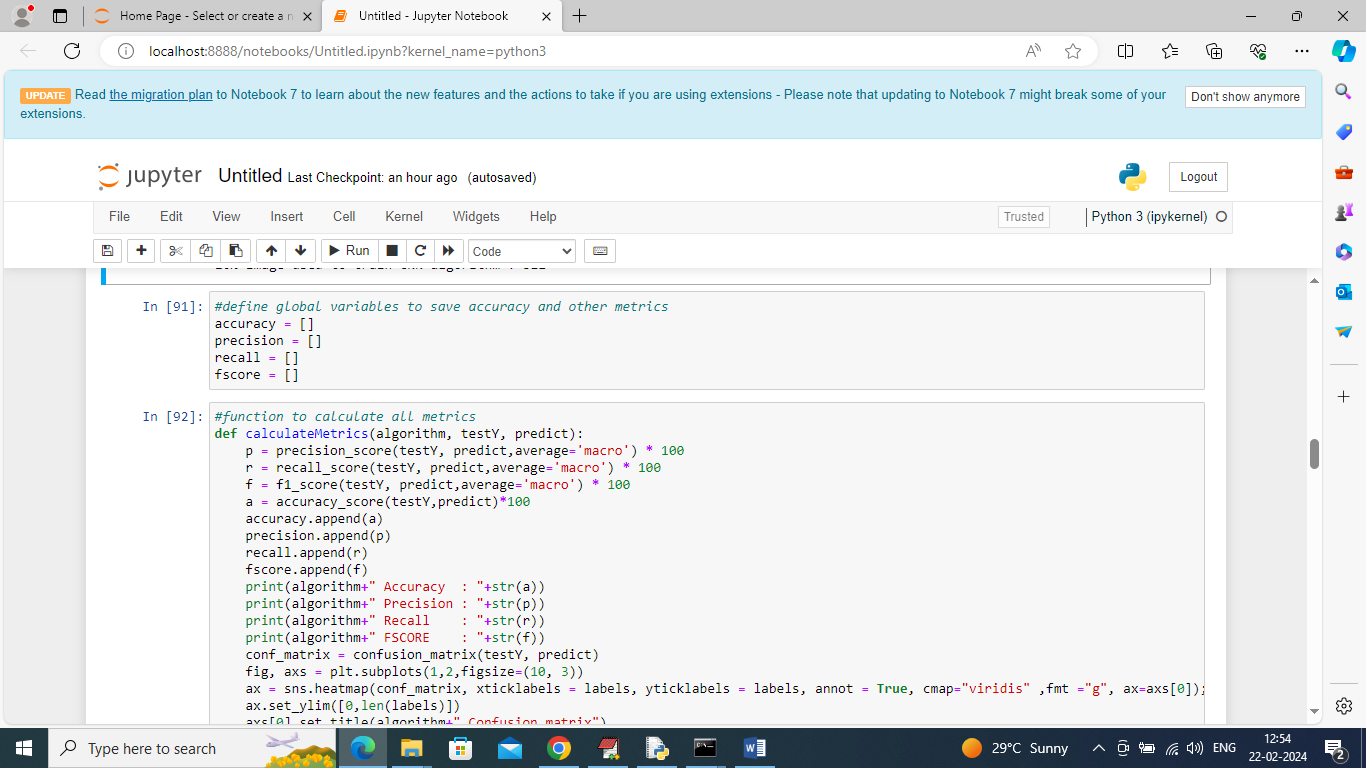
In above screen looping and reading all images from dataset folder and then resizing and adding to training X and Y array and then in blue colour text displaying total number of images loaded



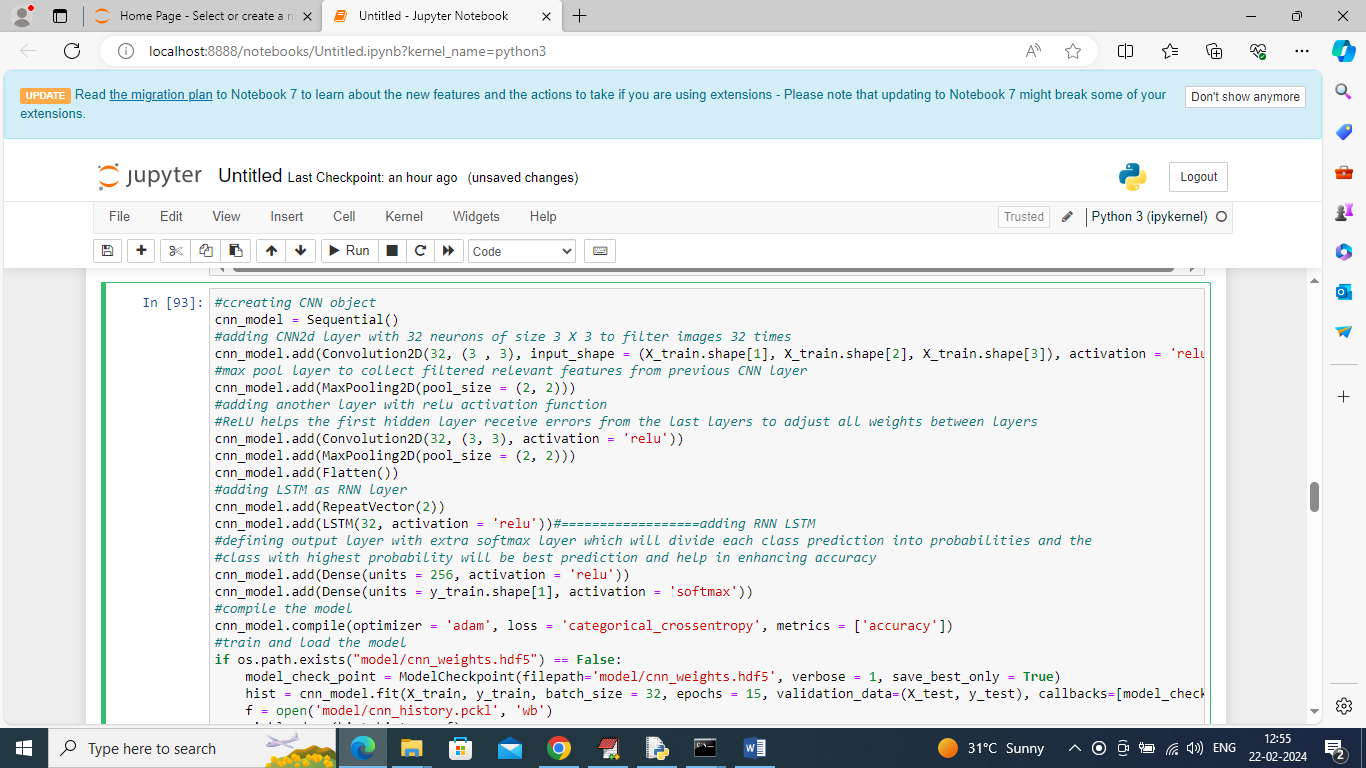
In above screen displaying graphs of different labels and number of images found in that label



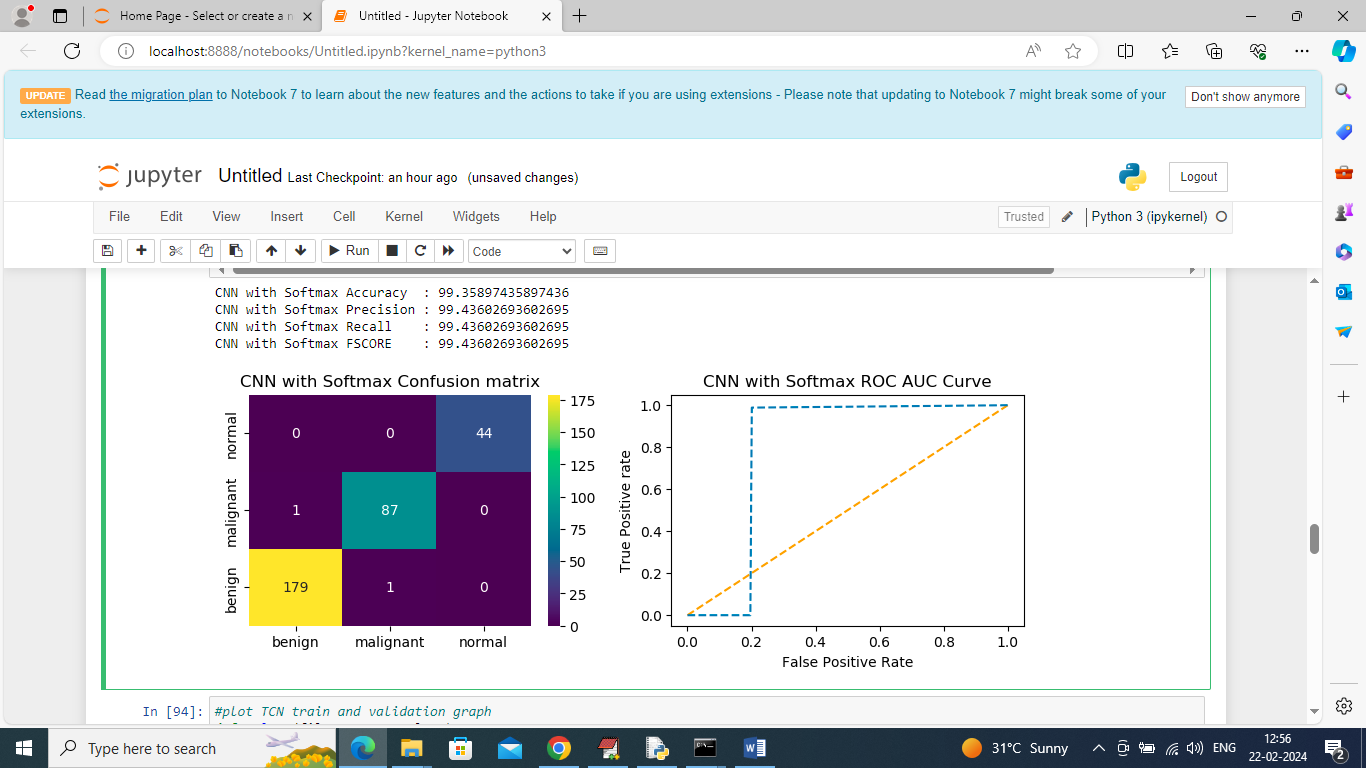
In above screen applying pre-processing techniques like Normalization, shuffling and splitting dataset into train and test where application using 80% dataset for training and 20% for testing and then in blue colour text displaying training and testing size images



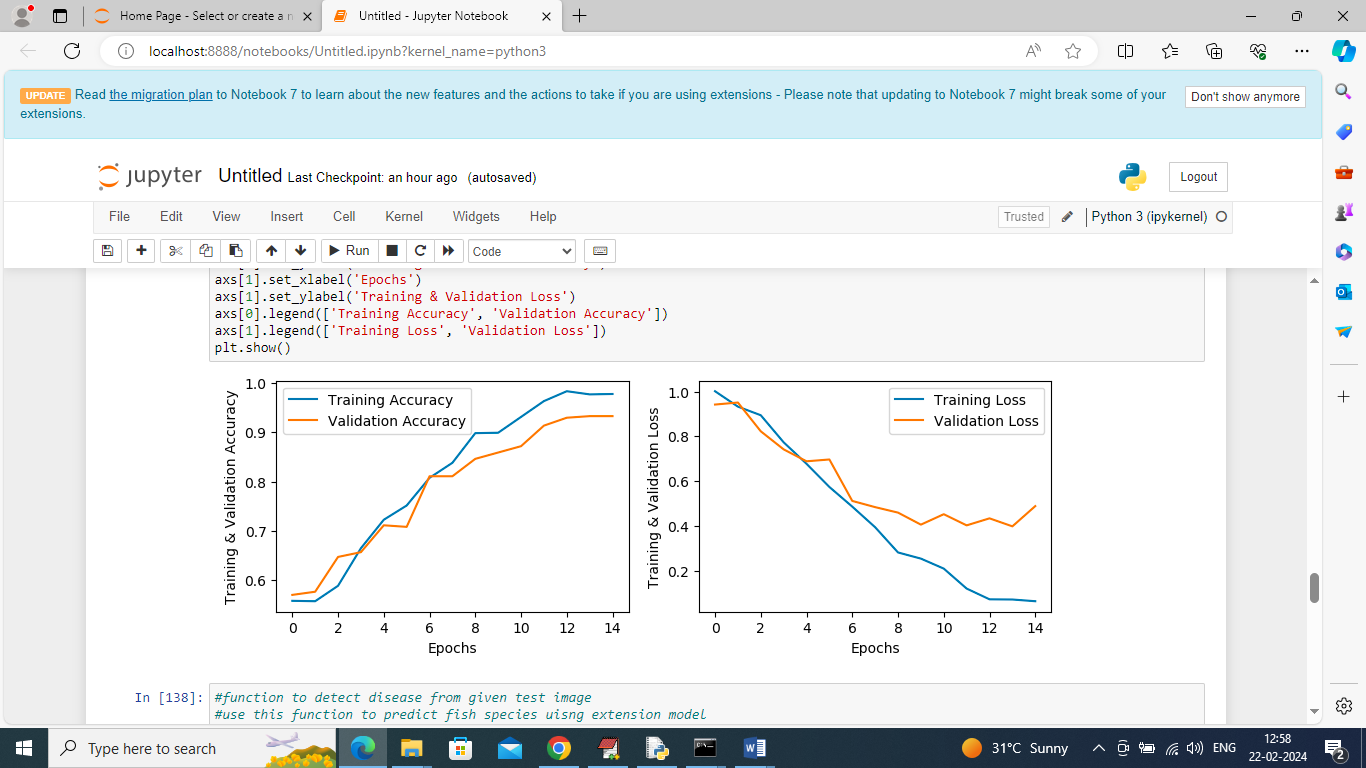
In above screen defining function to calculate accuracy and other metrics



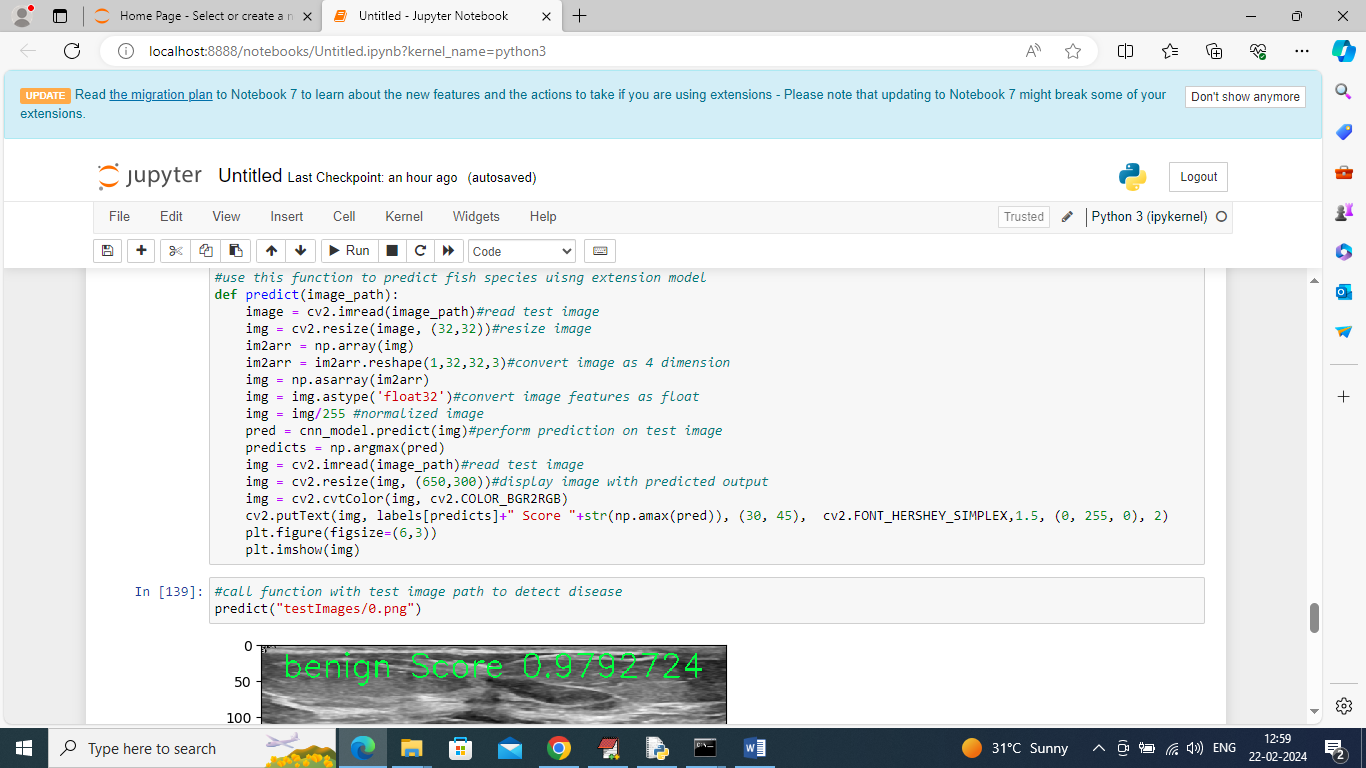
In above screen defining CNN and RNN LSTM layer for training and you can see LSTM layer at ‘======dashed’ lines so we are combining both CNN and LSTM as hybrid algorithm and after executing above block will get below output



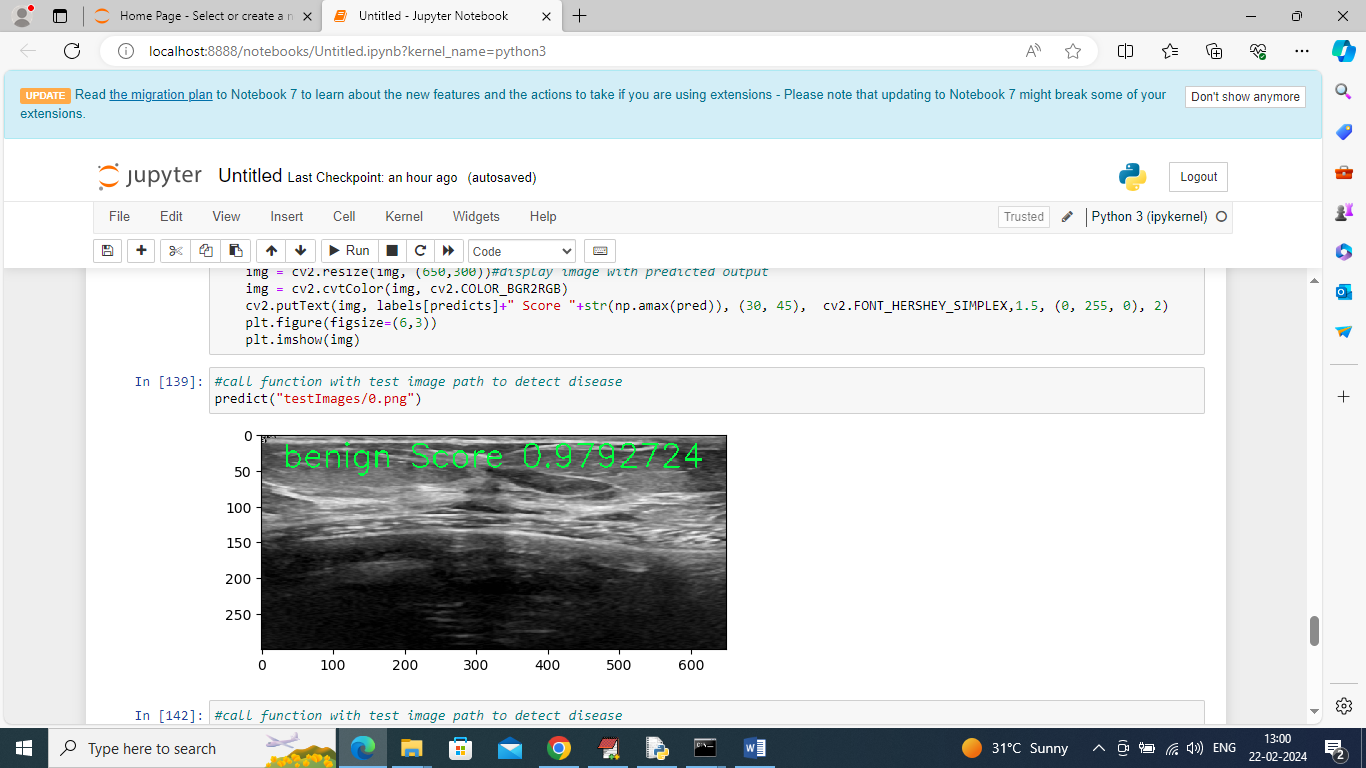
In above screen CNN with LSTM and Softmax got 99% accuracy and can see other metrics like precision, recall and etc. in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True labels and then different color boxes in diagnol represents correct prediction count and all blue boxes represents incorrect prediction count which are very few. In ROC graph x-axis represents False Positive Rate and y-axis represents True Positive Rate and if blue lines comes below orange line then all predictions are incorrect or false and if goes above orange line then all predictions are correct or true. In above ROC graph only few predictions are incorrect and maximum are correct



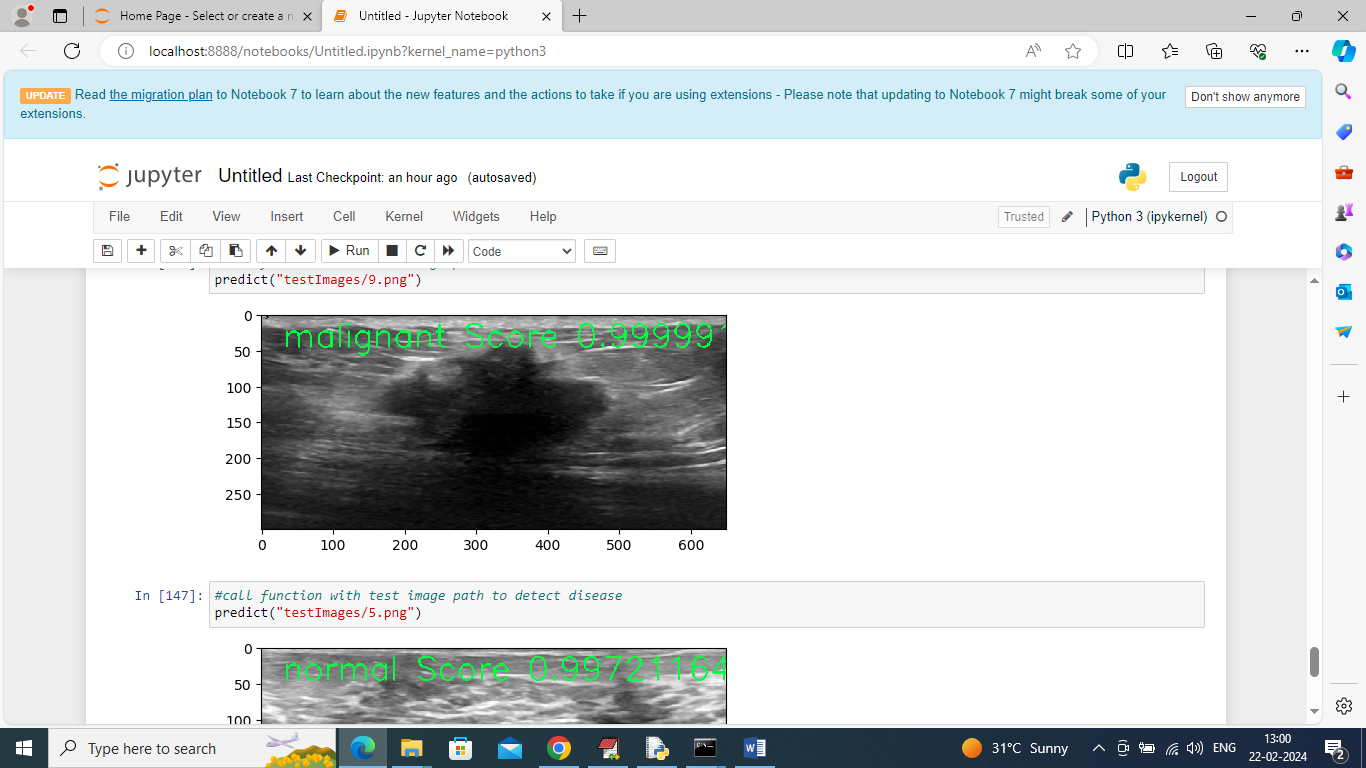
In above screen displaying CNN training and validation accuracy and loss values where x-axis represents training epochs and y-axis represents accuracy and loss in different lines and can see with each increasing epochs accuracy got increase and loss got decrease



In above screen defining function to read test image and then predict disease

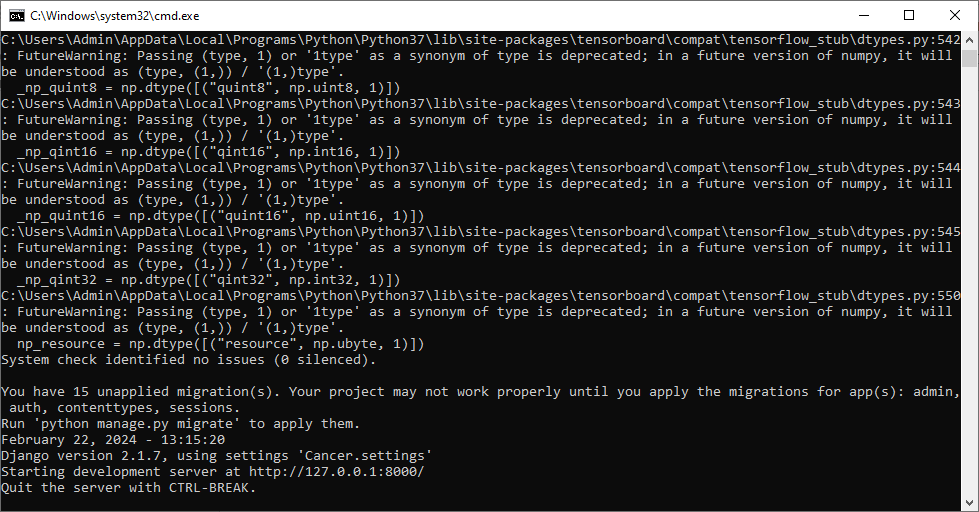


In above screen calling predict function with test image path and then in green color text can see predicted disease with score

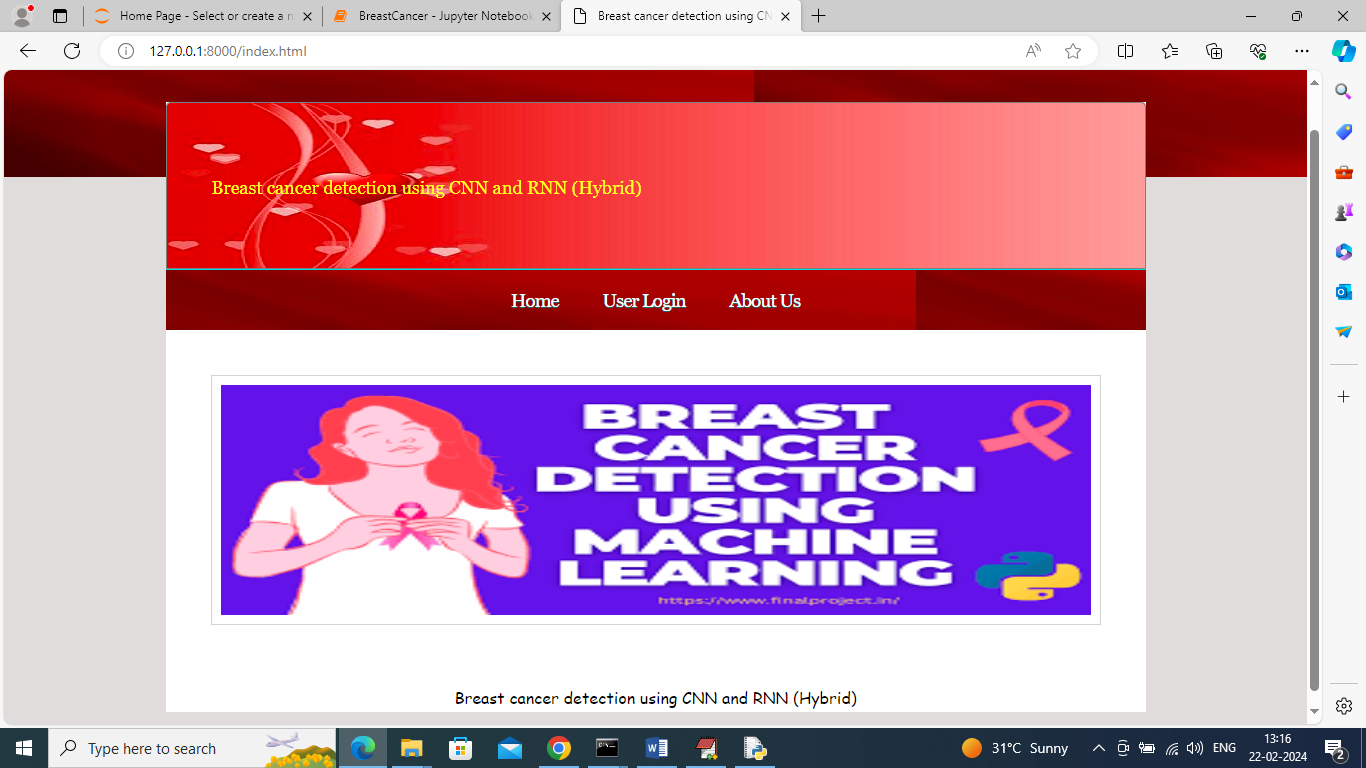


In above screen can see predictions from different images and same prediction we can see from below web output.

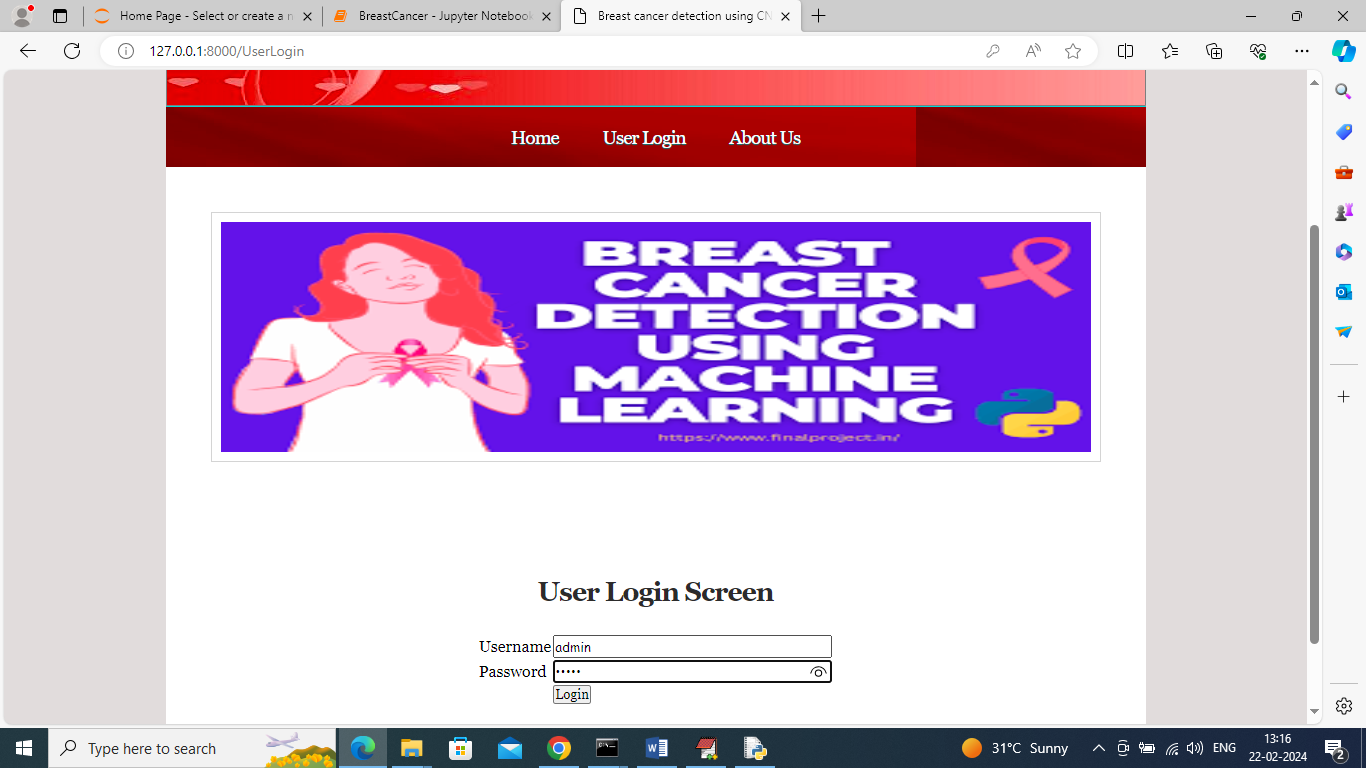
To run web code double click on ‘runServer.bat’ file to start python web server and get below page



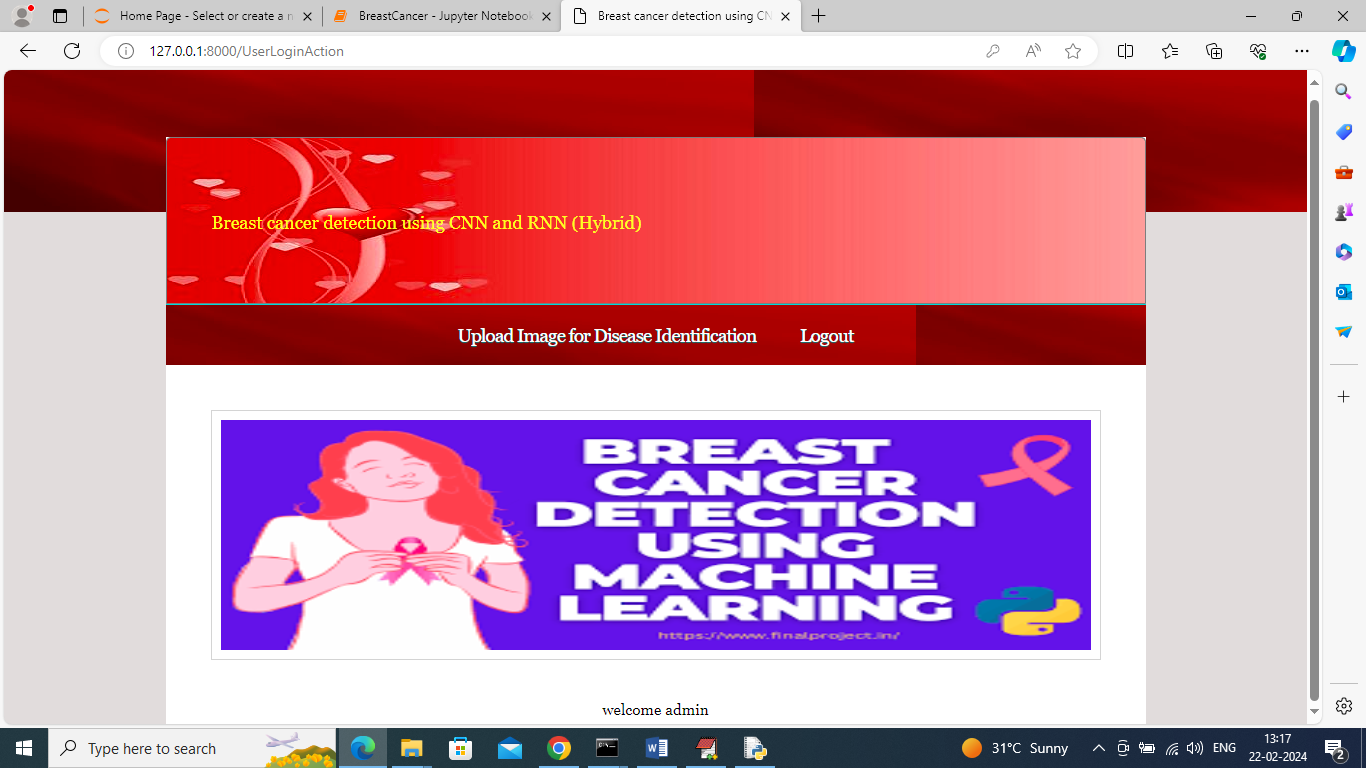
In above screen python server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below page



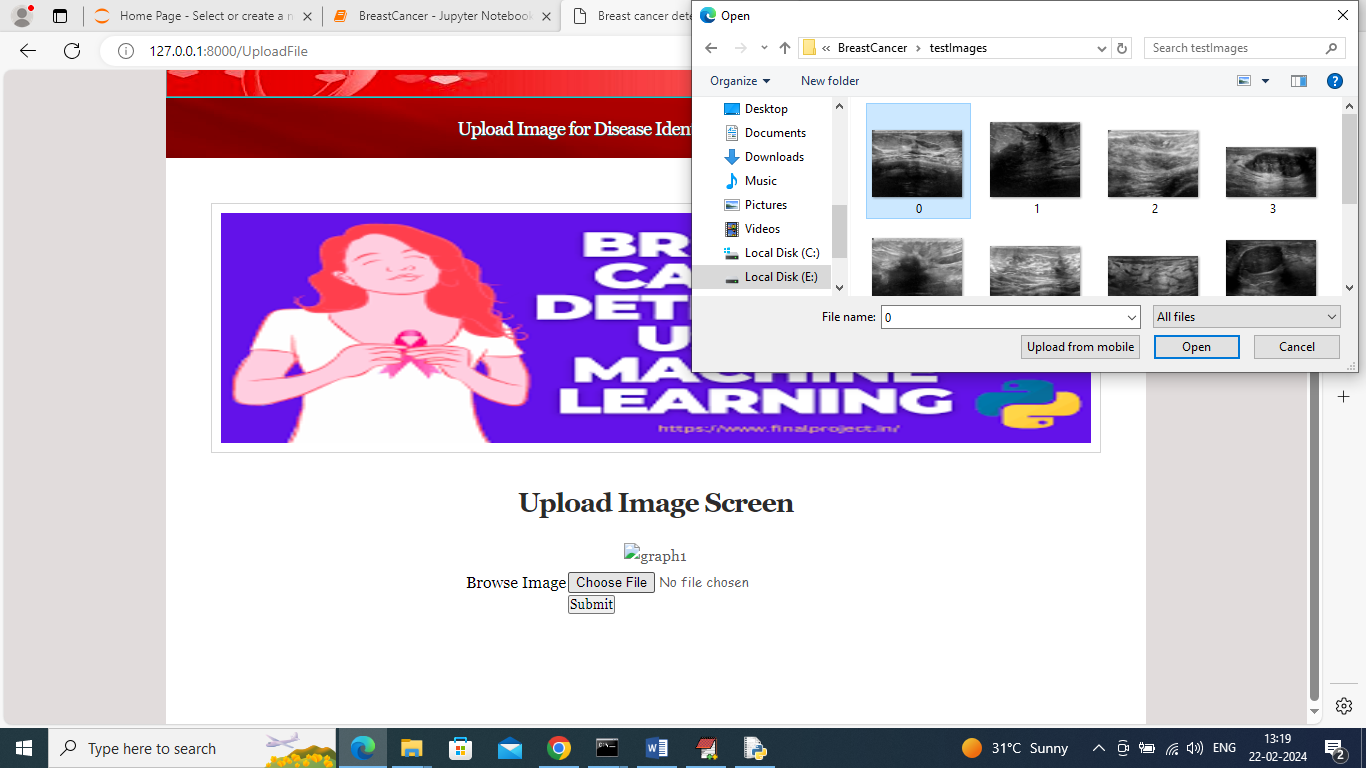
In above screen click on ‘User Login’ link to get below page



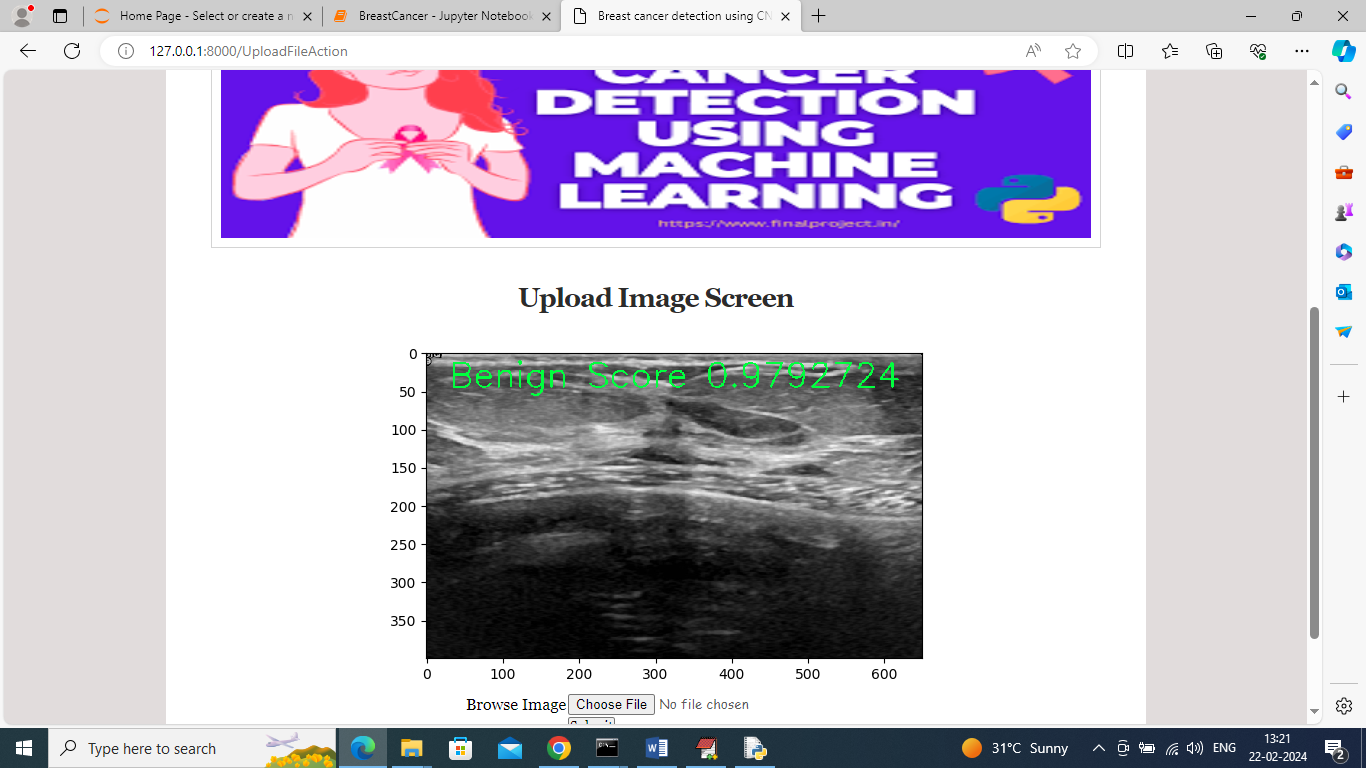
In above screen user can login using username and password as ‘admin and admin’ and then click on ‘Login’ button to get below page



In above screen click on ‘Upload Image for Disease Identification’ link to upload image



In above screen selecting and uploading test image and then click on ‘Open’ and ‘submit’ button to get below page



In above screen can see detected disease printed on image and similarly you can upload and test other images and below is another output

