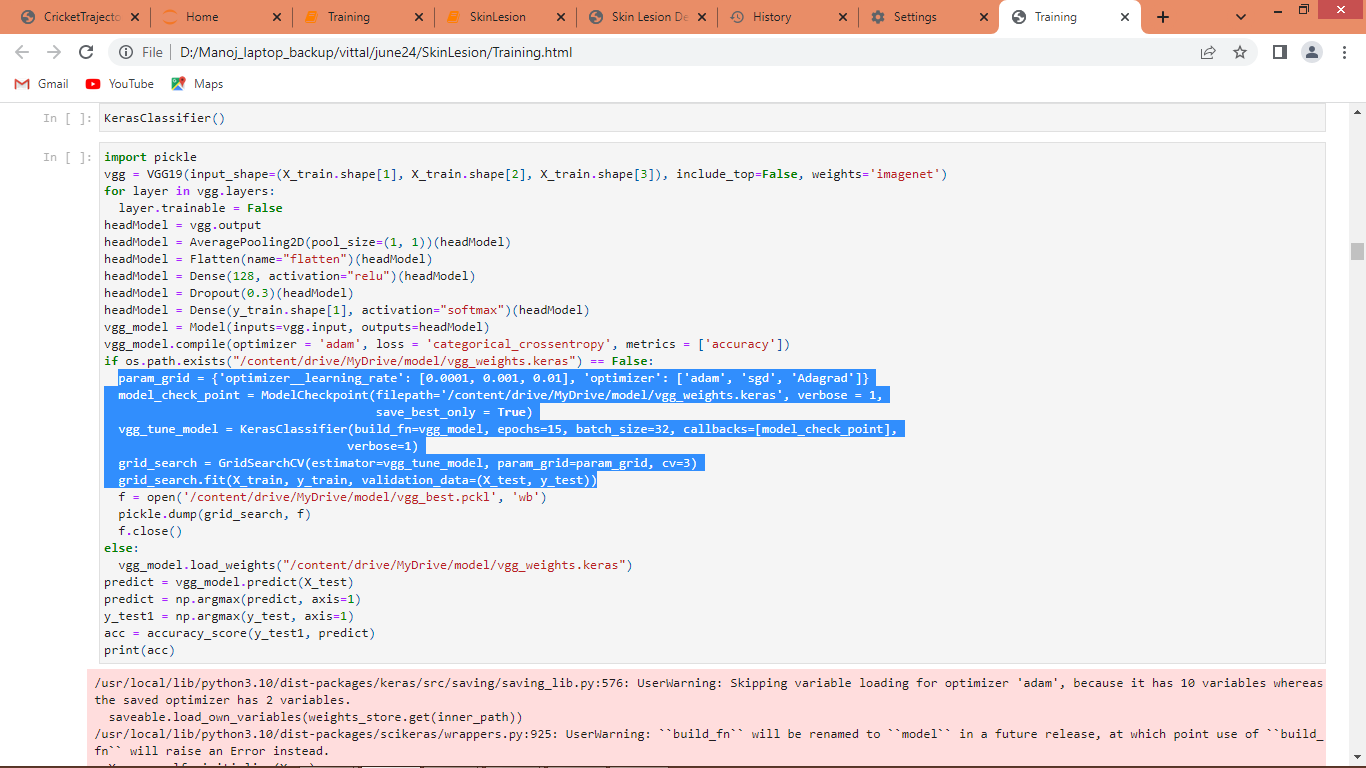
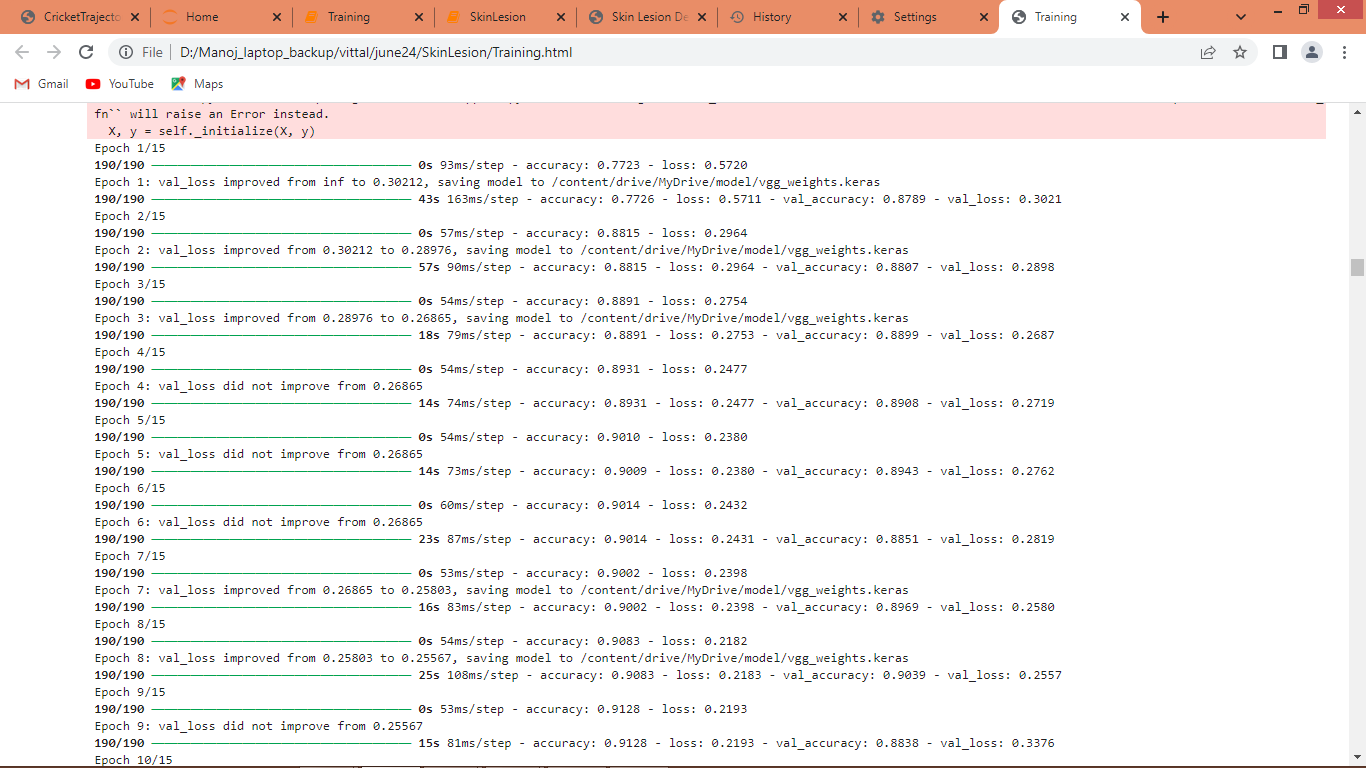
Dermatological Image Classification for Skin Lesion Detection using Deep Learning Techniques

In this project you requirement is to train pre-trained model such Xception, VGG19 and ALEXNET to detect Skin Lesion. To train these models we have use tuning functionalities with different optimizers and learning rate and to tune algorithm we have Grid Search algorithm. Each algorithm performance is evaluated in terms of accuracy, precision, recall, FSCORE, confusion matrix and ROC graph. Among all algorithms VGG19 giving best accuracy closer to 92%.

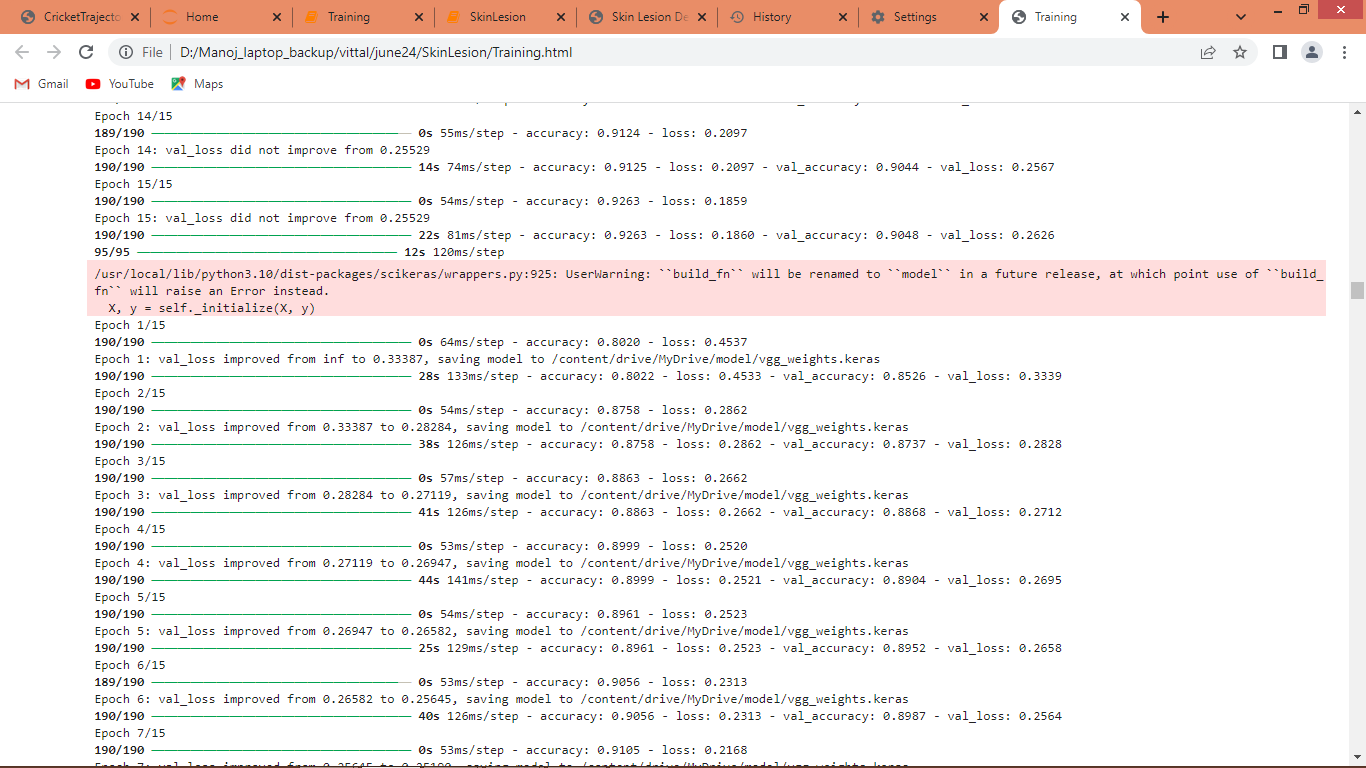
To train above models we have used same dataset given by you and for training we have used Google COLAB as system was getting crash because of heavy training with different tuning parameters. In below screen showing COLAB code training with parameters



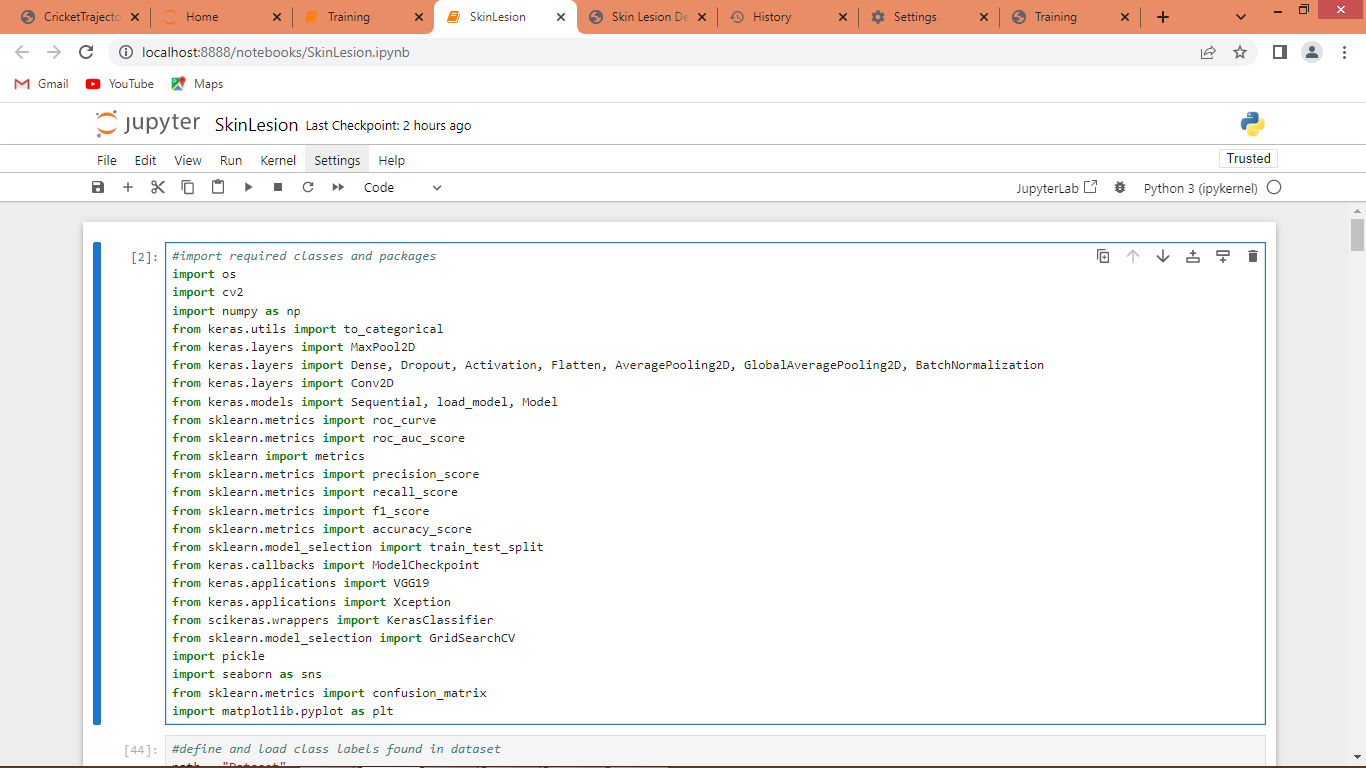
In above screen you can see we are training VGG19 as transfer learning algorithm and then in blue colour text can see definition of tuning parameters with different optimizers and learning rate and in below screen can see training of algorithm epoch by epoch



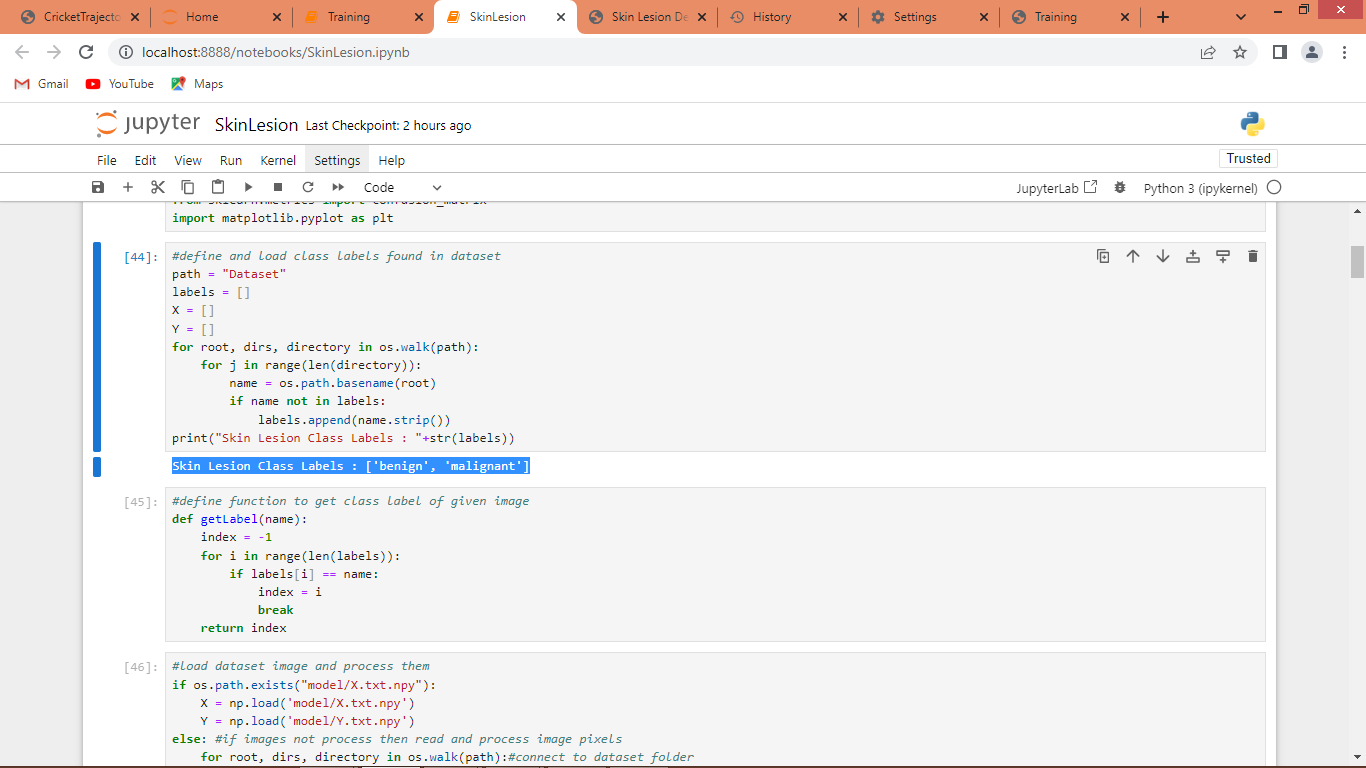
In below screen can see same algorithm training with other parameters and this continues till all tuning parameters applied



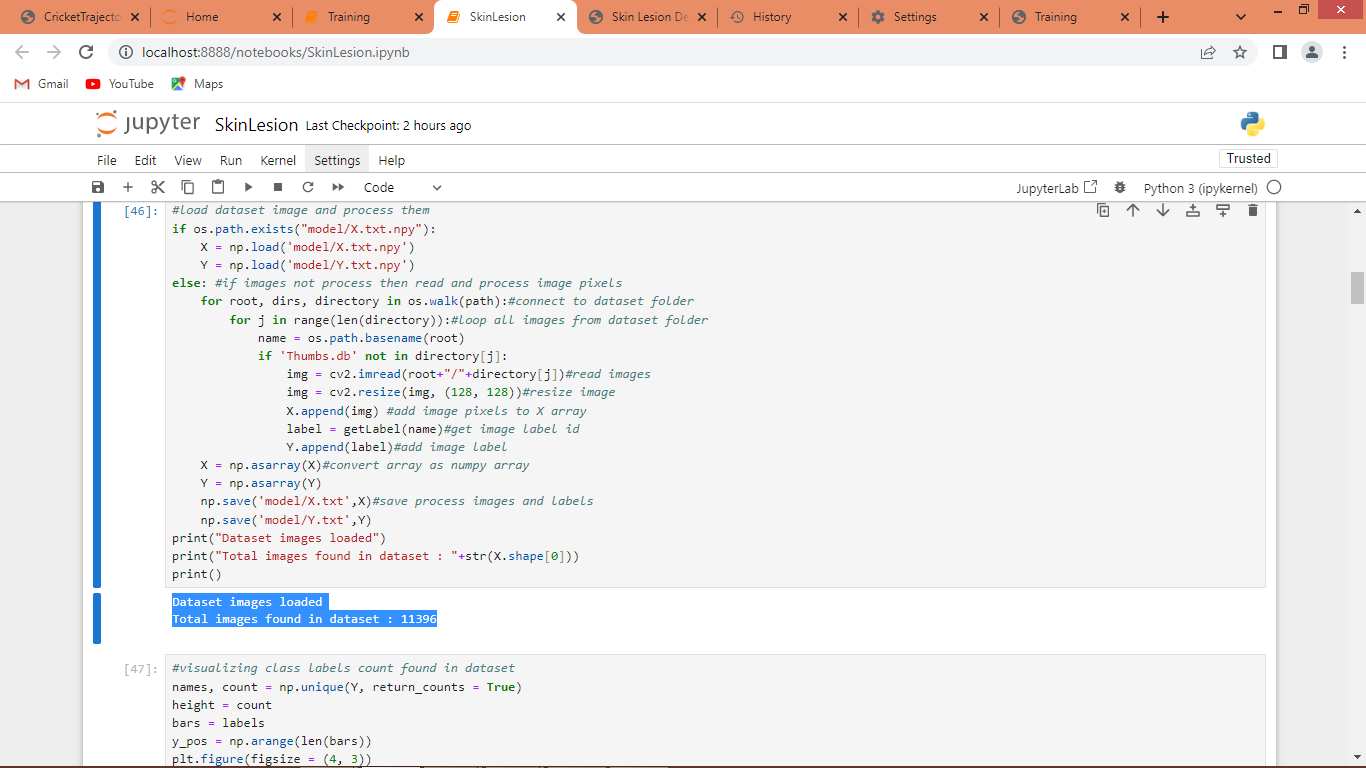
In COLAB to train all algorithms it took more than 12 hours and in below screen showing JUPYTER code with all steps and each JUPYTER block is marked with blue colour comments



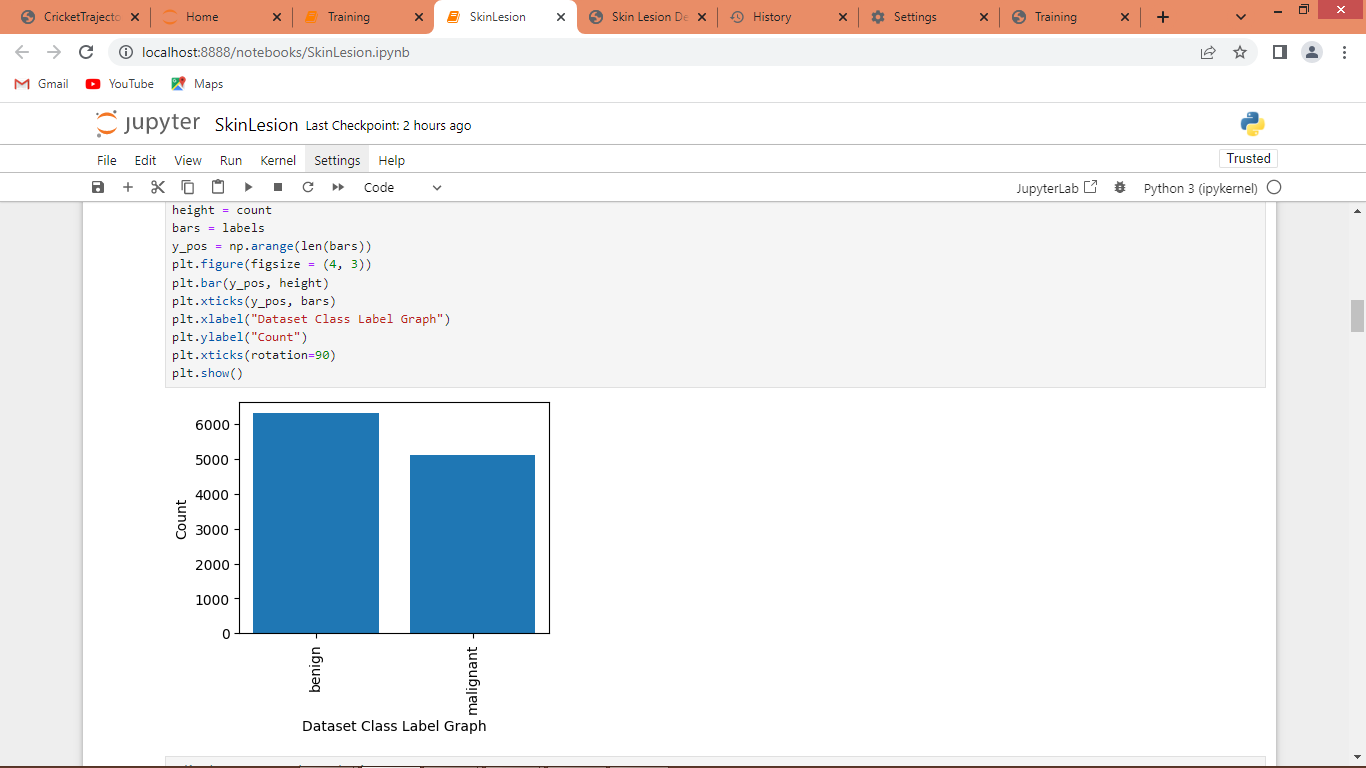
In above screen loading and importing required python classes and packages



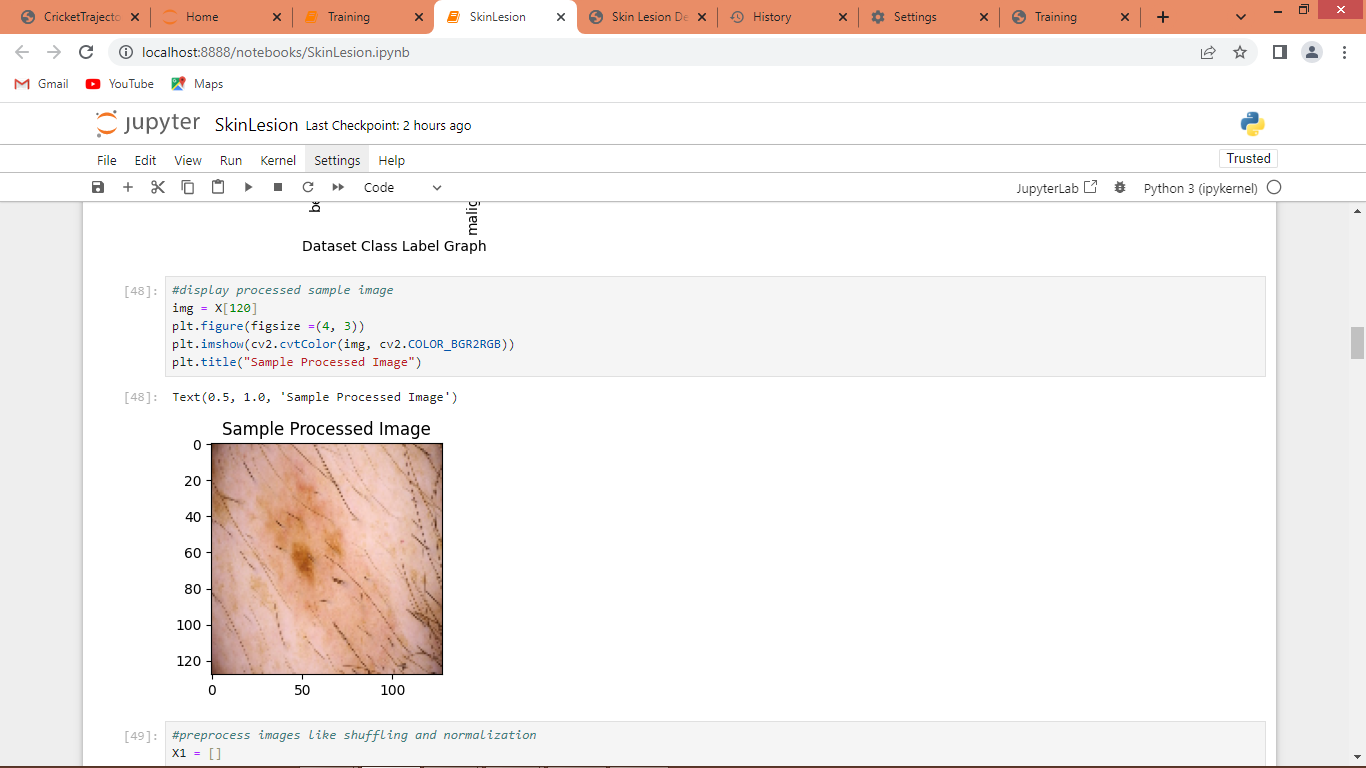
In above screen connecting to dataset folder and then in blue colour text displaying different class labels found in dataset



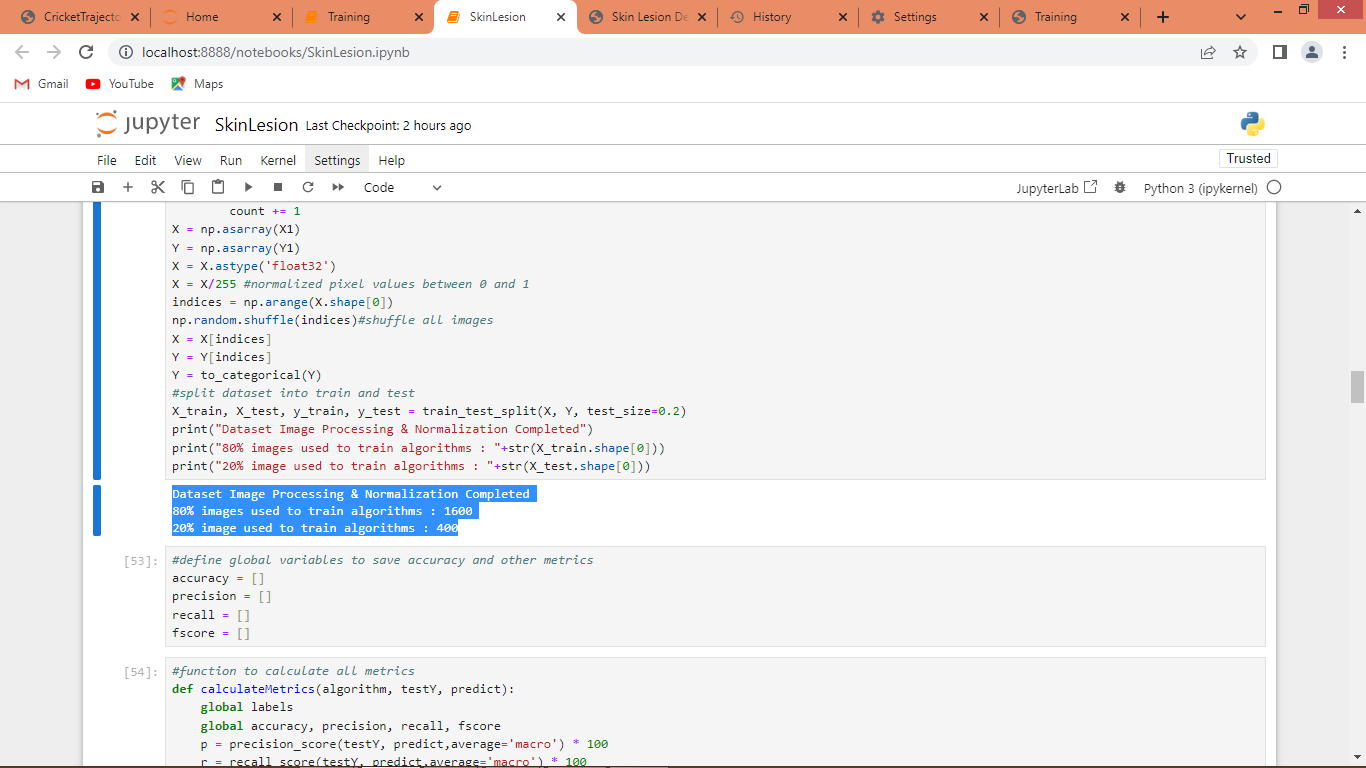
In above screen from dataset folder reading and processing each image and then in blue colour text can see total 11000 images loaded



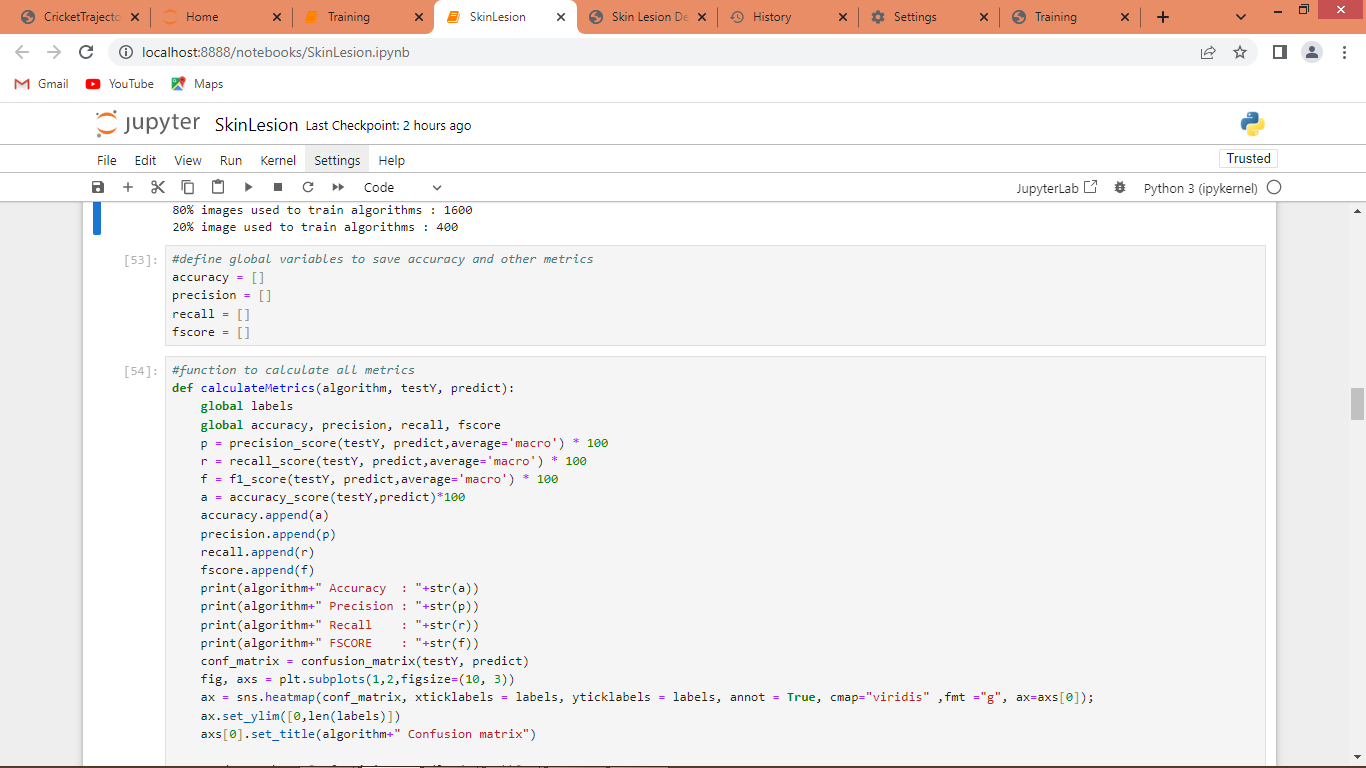
In above screen visualizing graph of different class labels and number of images exists under each class labels



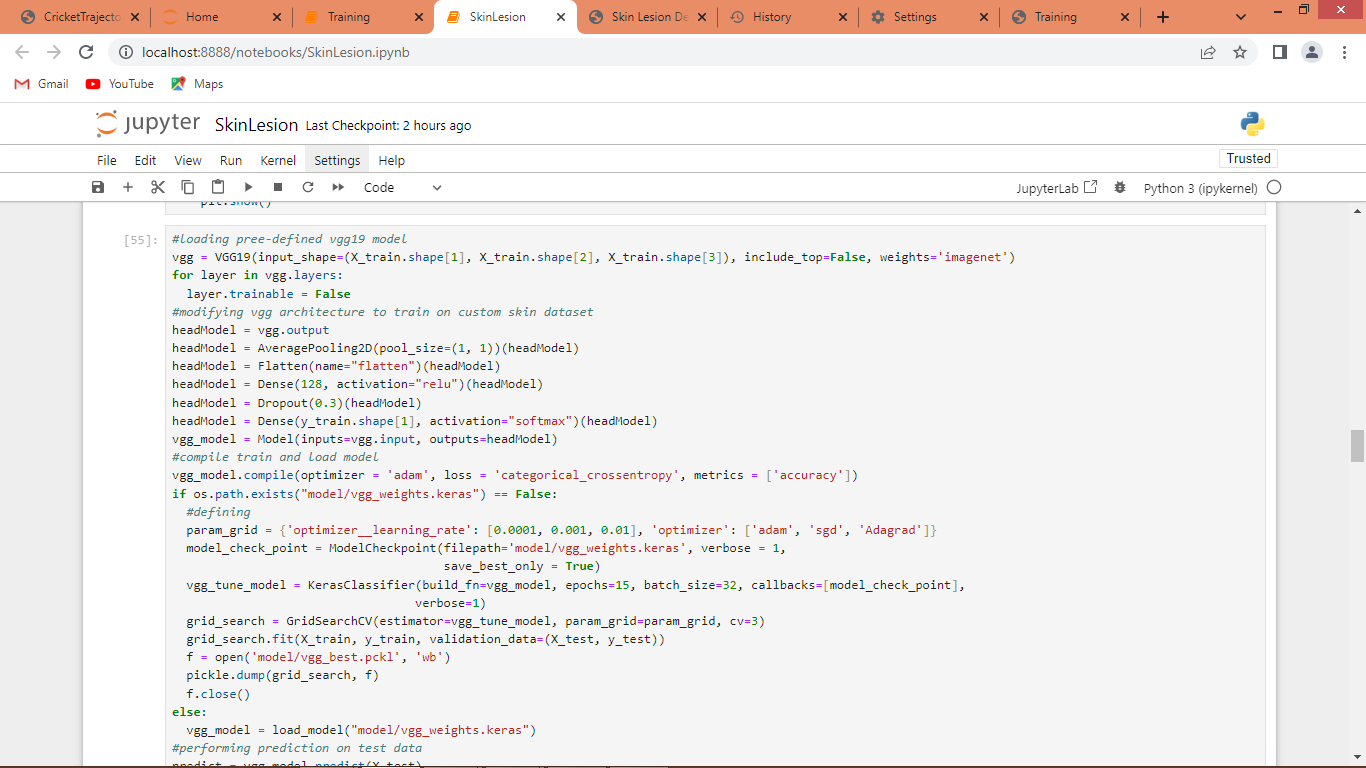
In above screen displaying sample loaded images



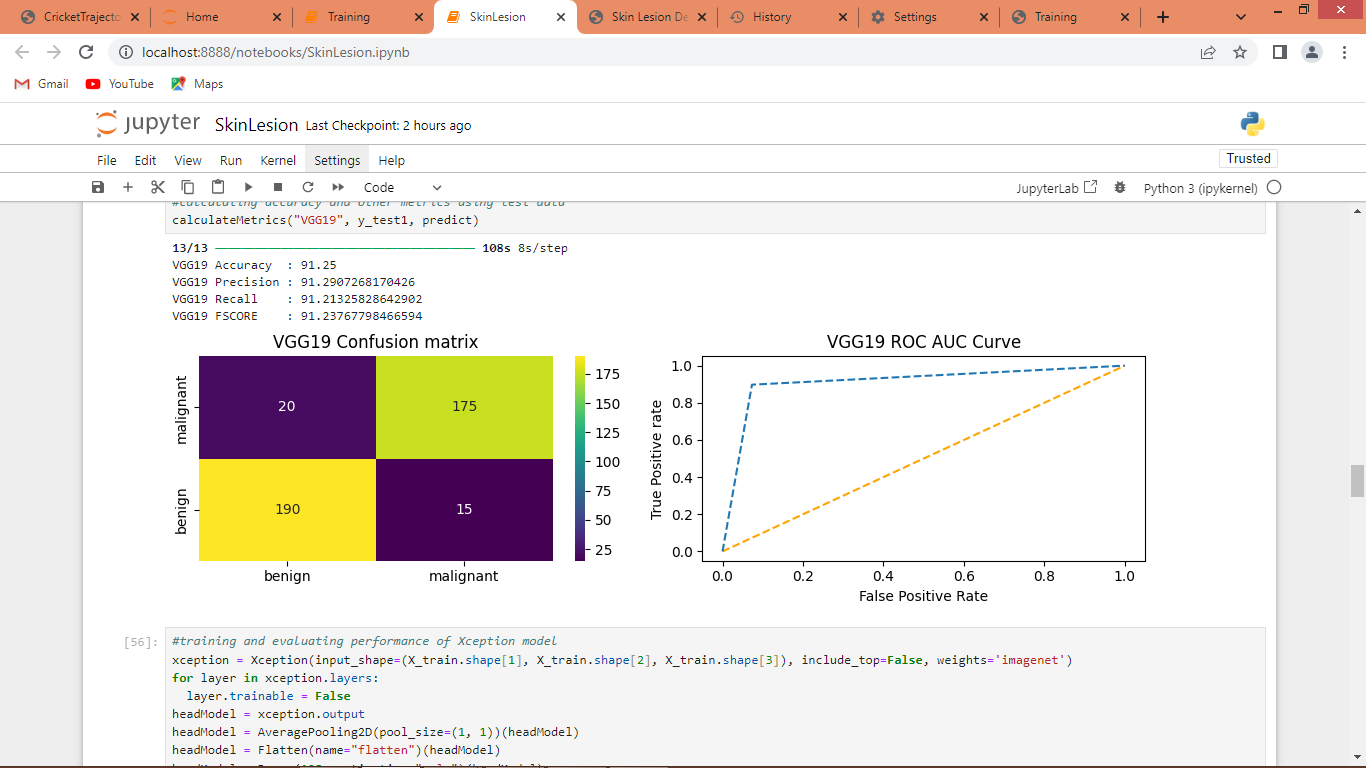
In above screen applying processing techniques such as shuffling, normalization and then splitting dataset into train and test where application using 80% images for training and 20% for testing and in blue colour text can see train and test size



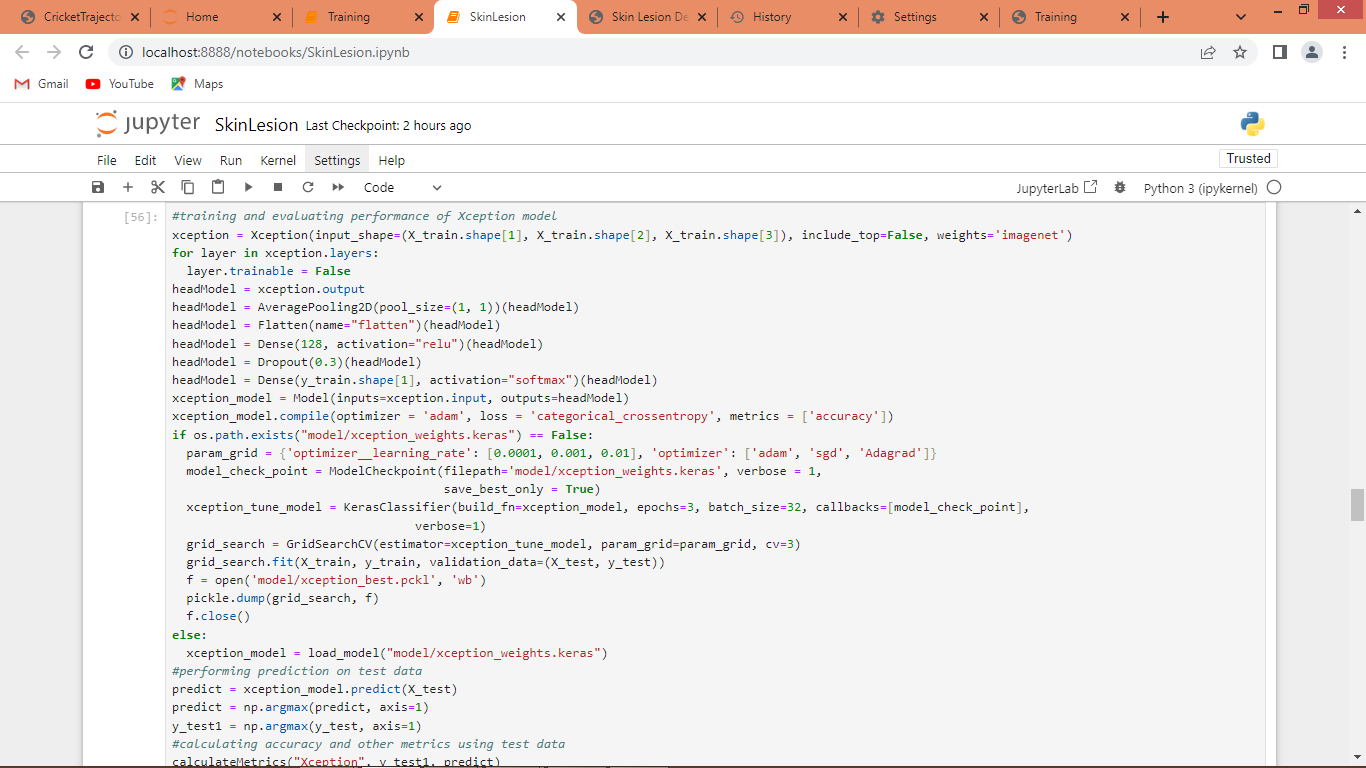
In above screen defining function to calculate accuracy and other metrics



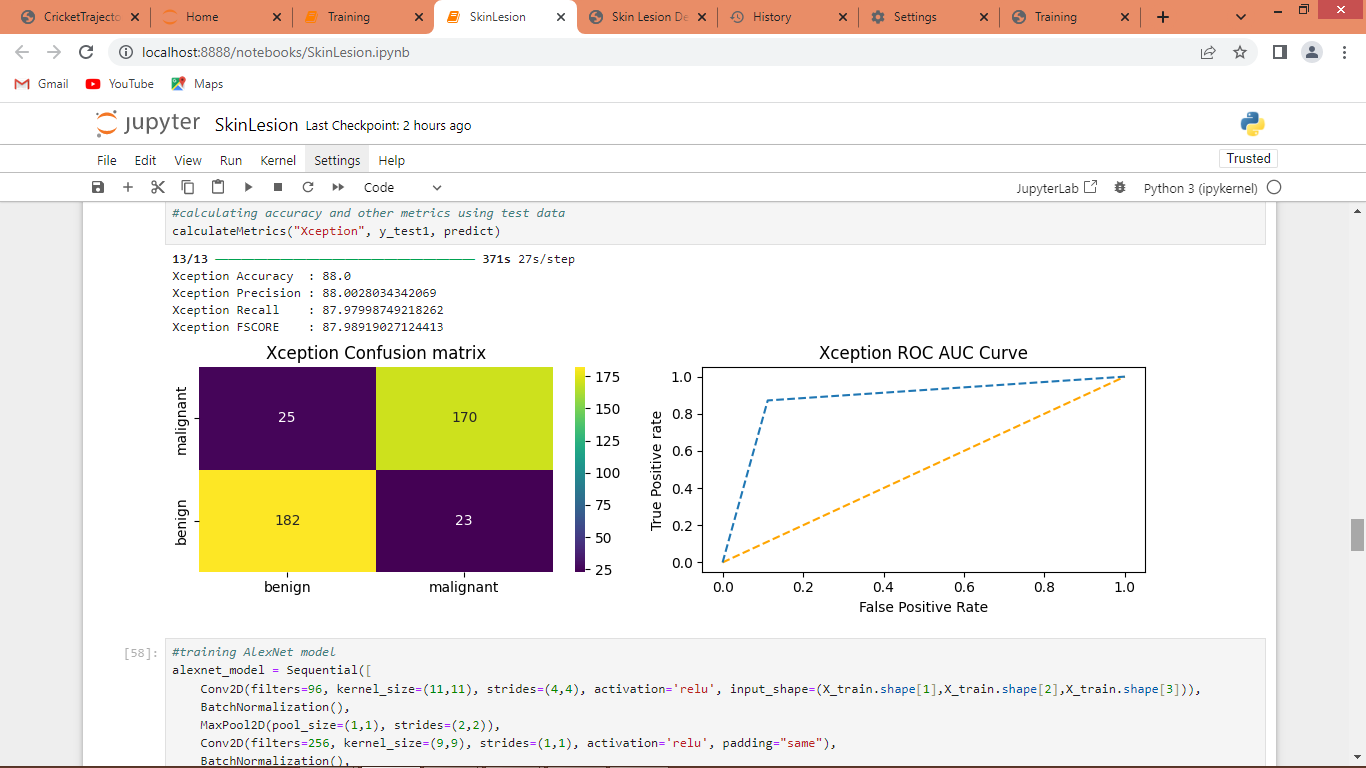
In above screen training VGG19 algorithm and after training will get below output



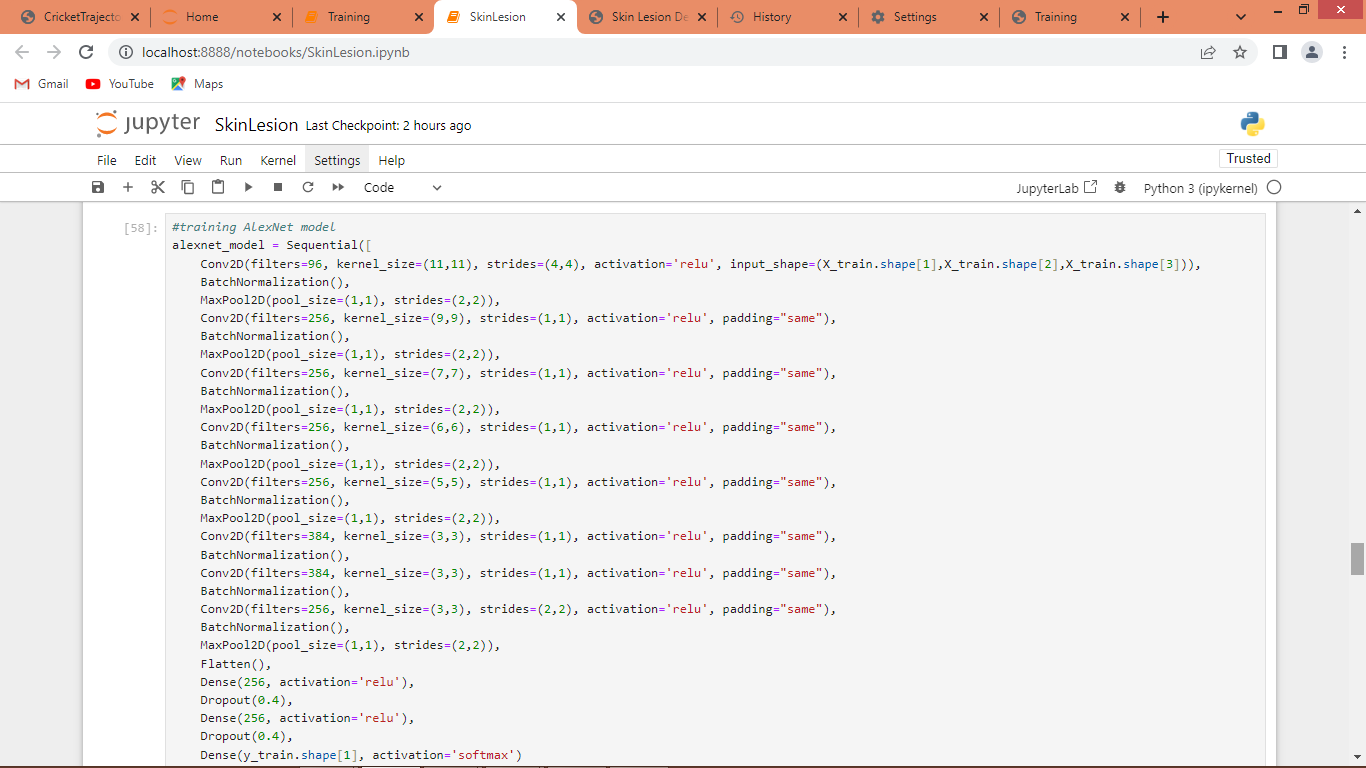
In above screen VGG19 got 91% accuracy and can see other metrics like precision, recall, FSCORE. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and then all yellow colour boxes represents correct prediction count and blue boxes represents incorrect prediction count which are very few. In ROC graph x-axis represents False Positive Rate and y-axis represents True Positive Rate and if blue line comes on top of orange line then all predictions are correct and if goes below orange line then all predictions are incorrect.



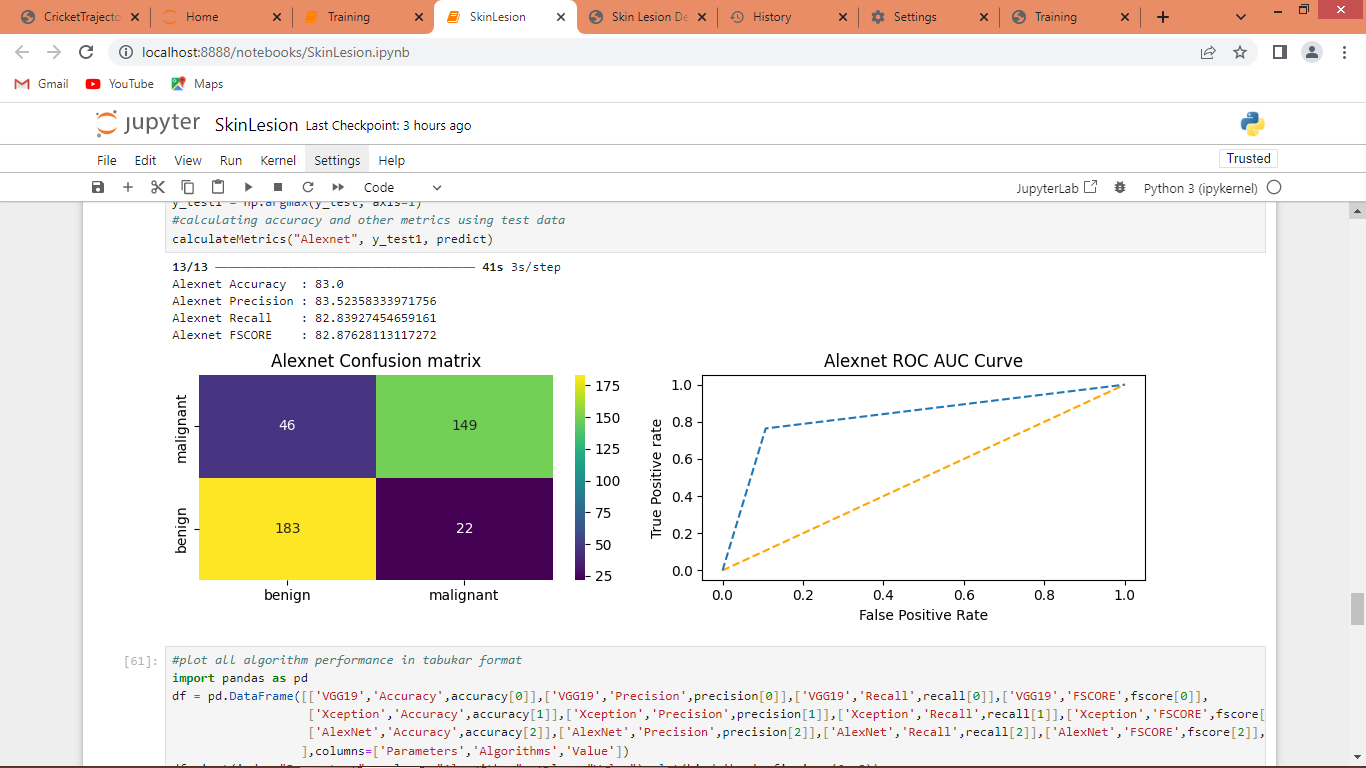
In above screen training and loading XCEPTION model and below is the output



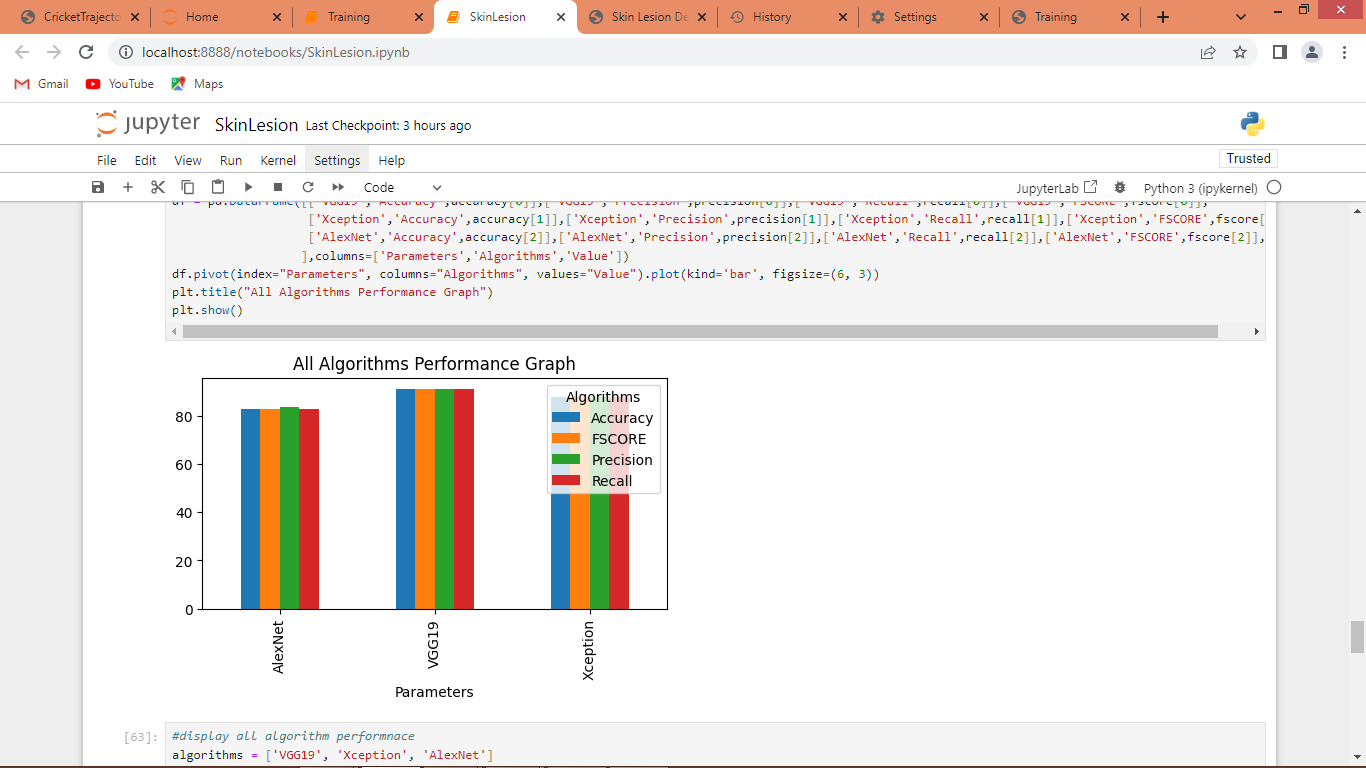
In above screen XCEPTION got 88% accuracy and can see other metrics also



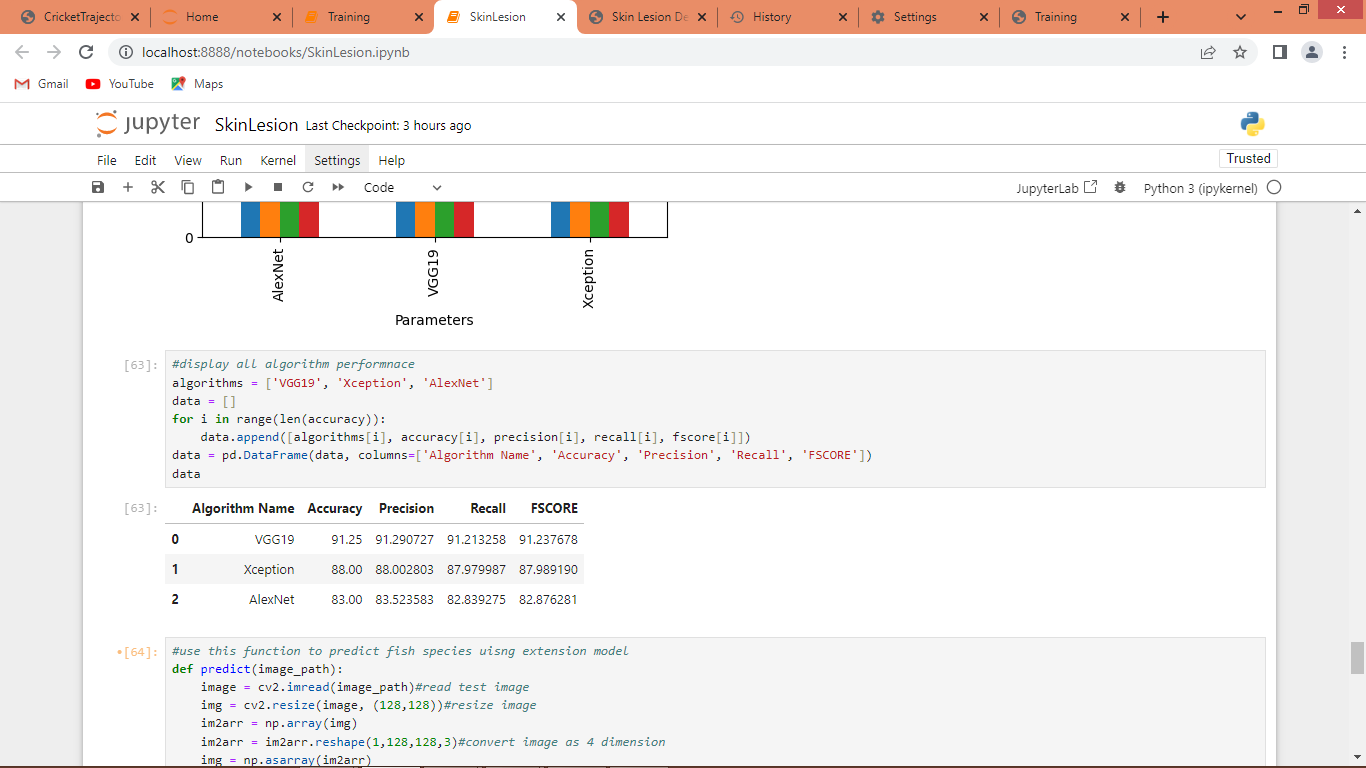
In above screen training and loading ALEXNET model and below is the output



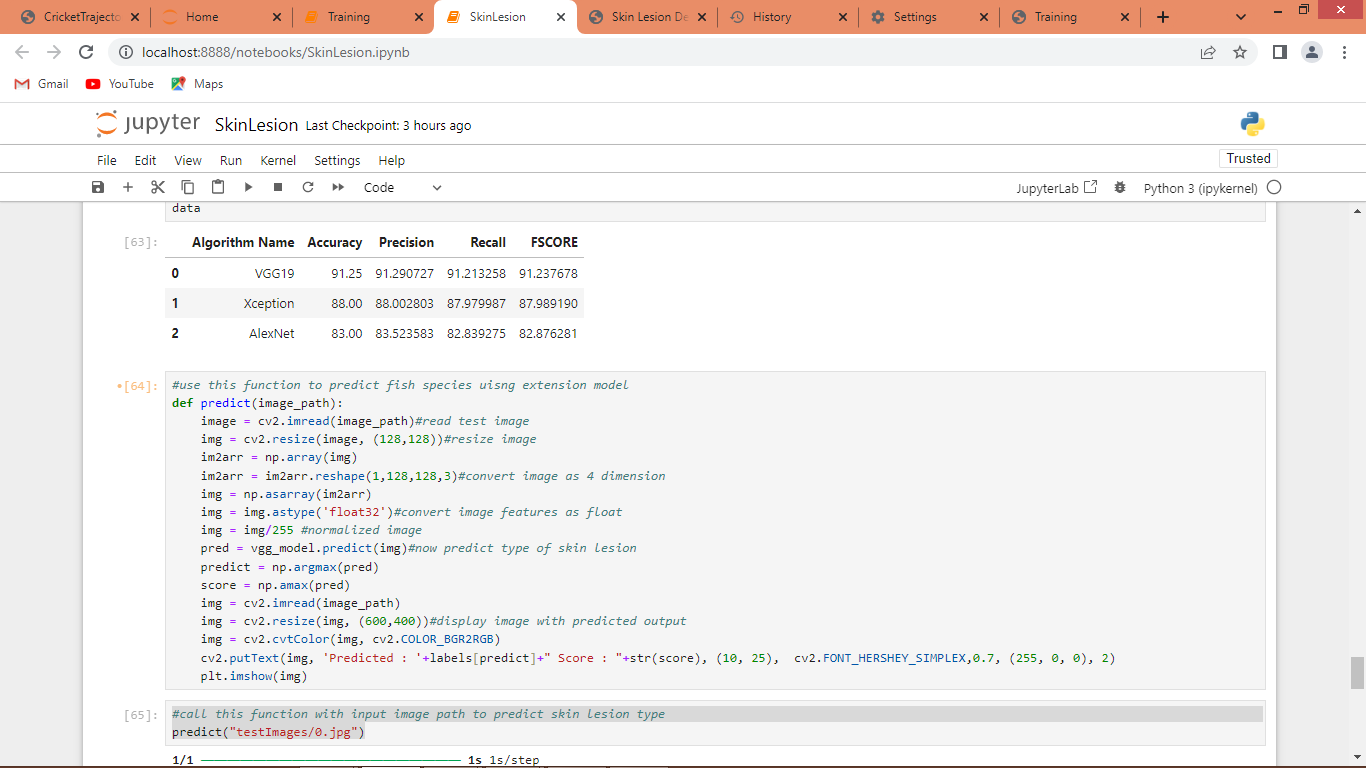
In above screen ALEXNET model got 83% accuracy



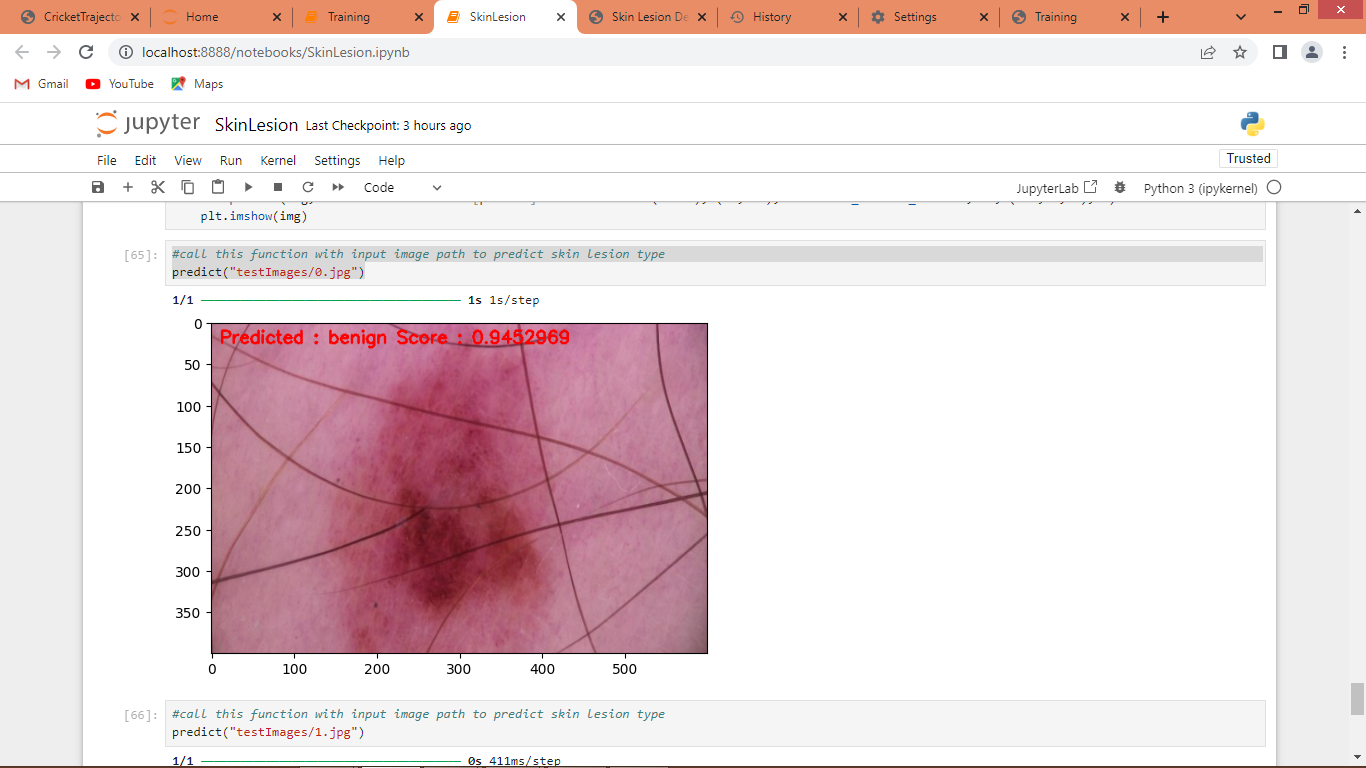
In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms VGG got high accuracy



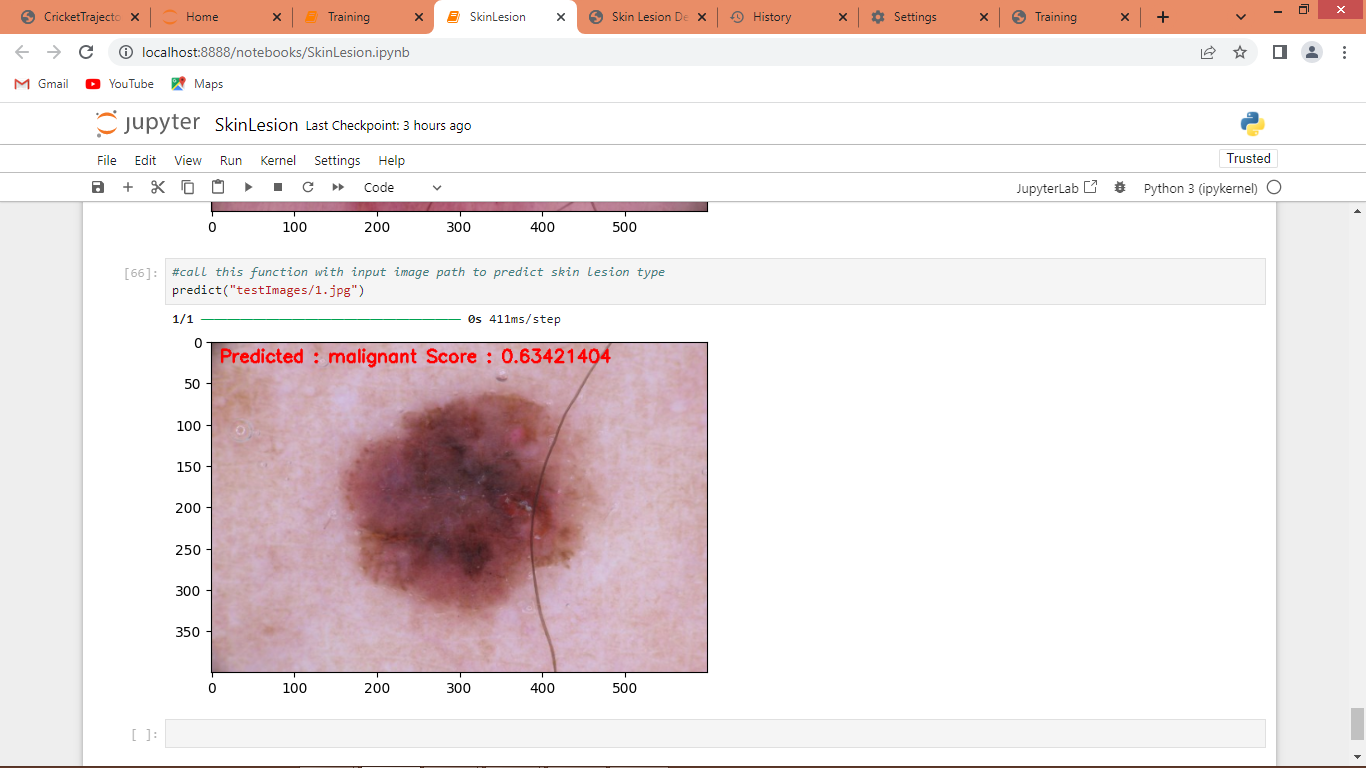
In above screen can see all algorithm performance in tabular format



In above screen defining predict function which will take image path as input and then apply VGG model to predict skin lesion

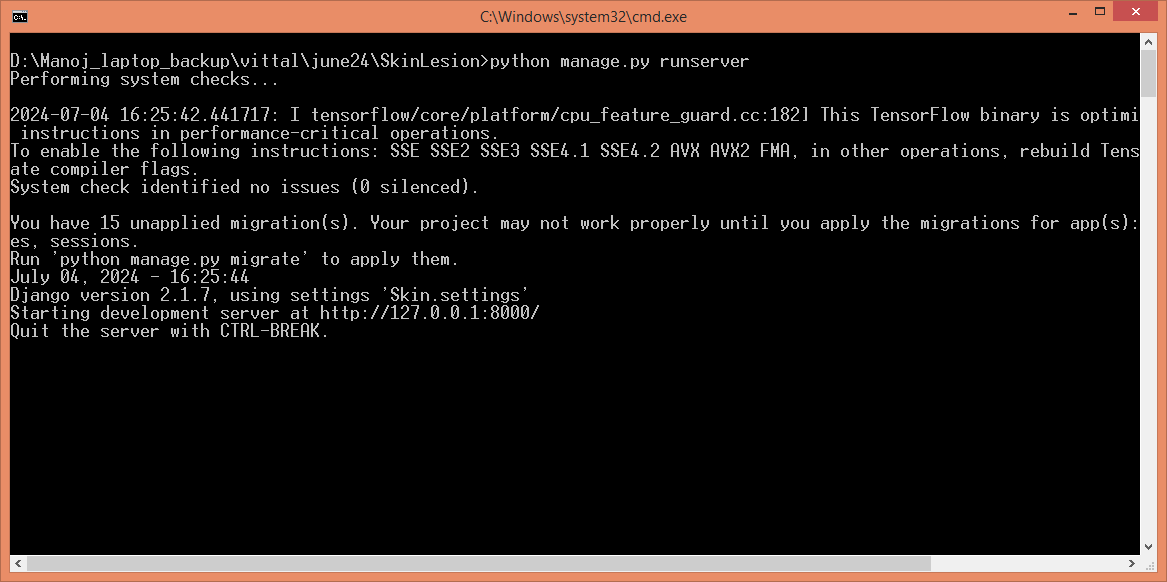


In above screen calling predict function with image path and then skin lesion predicted as Benign with a score of 94%

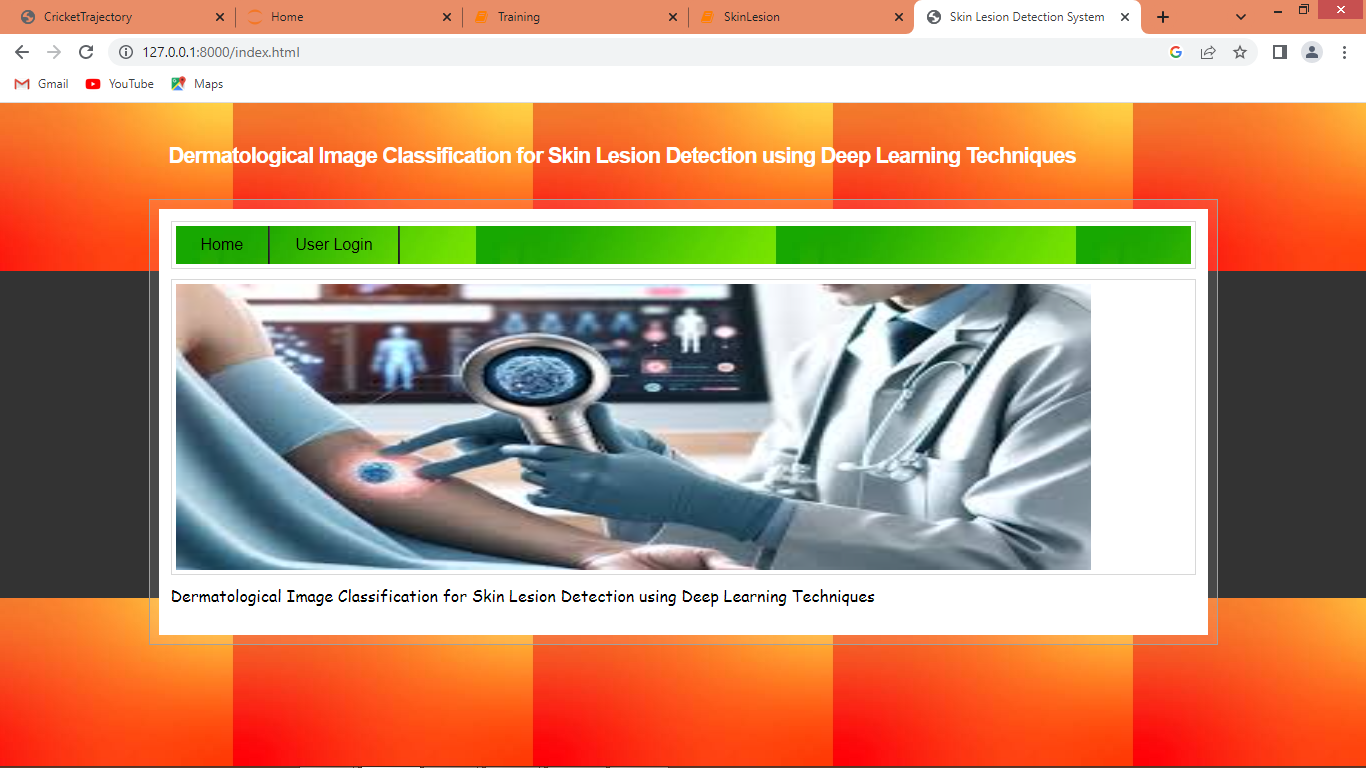


In above screen another image predicted as Malignant and similarly you can call this function with different image path to get skin lesion. All testing images i saved inside ‘testImages’ folder.

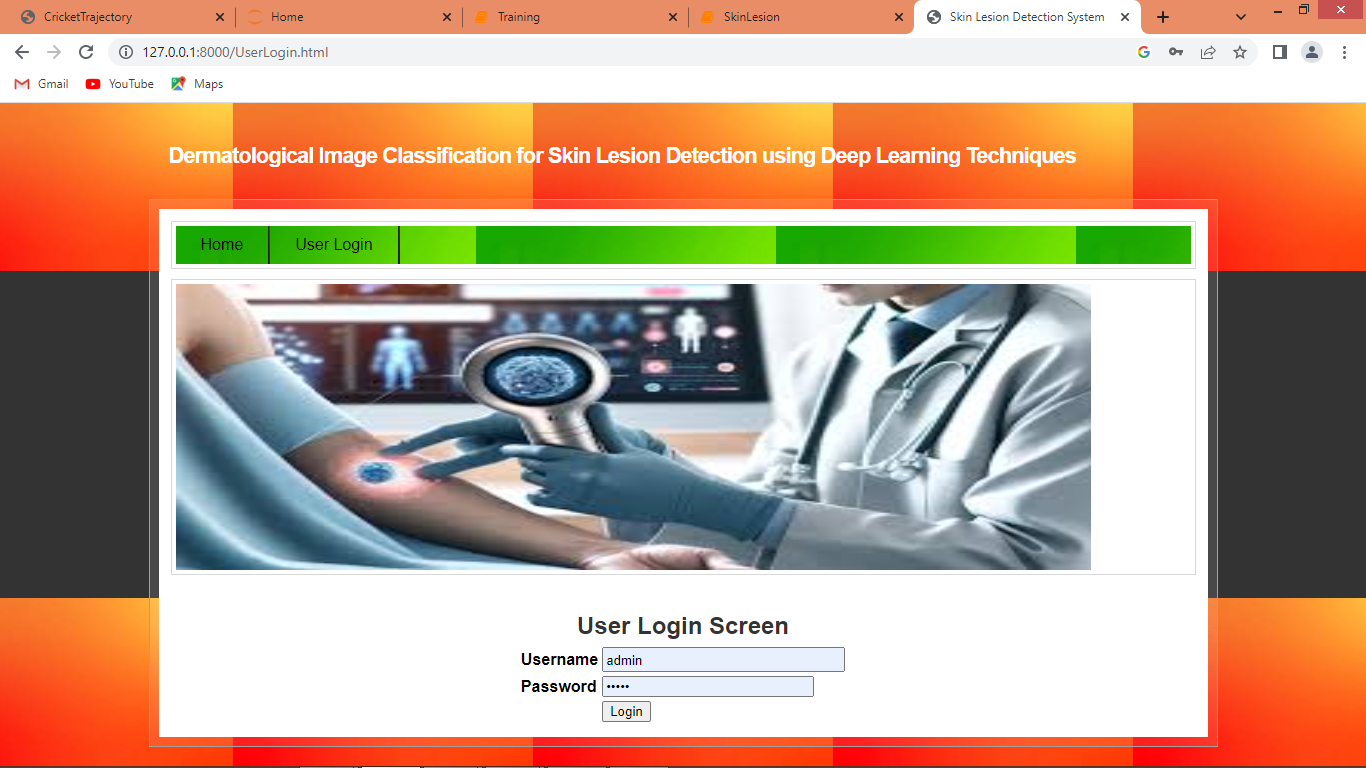
Now same code we can run from WEB also by double click on ‘runWEB.bat’ file to start web server and get below page



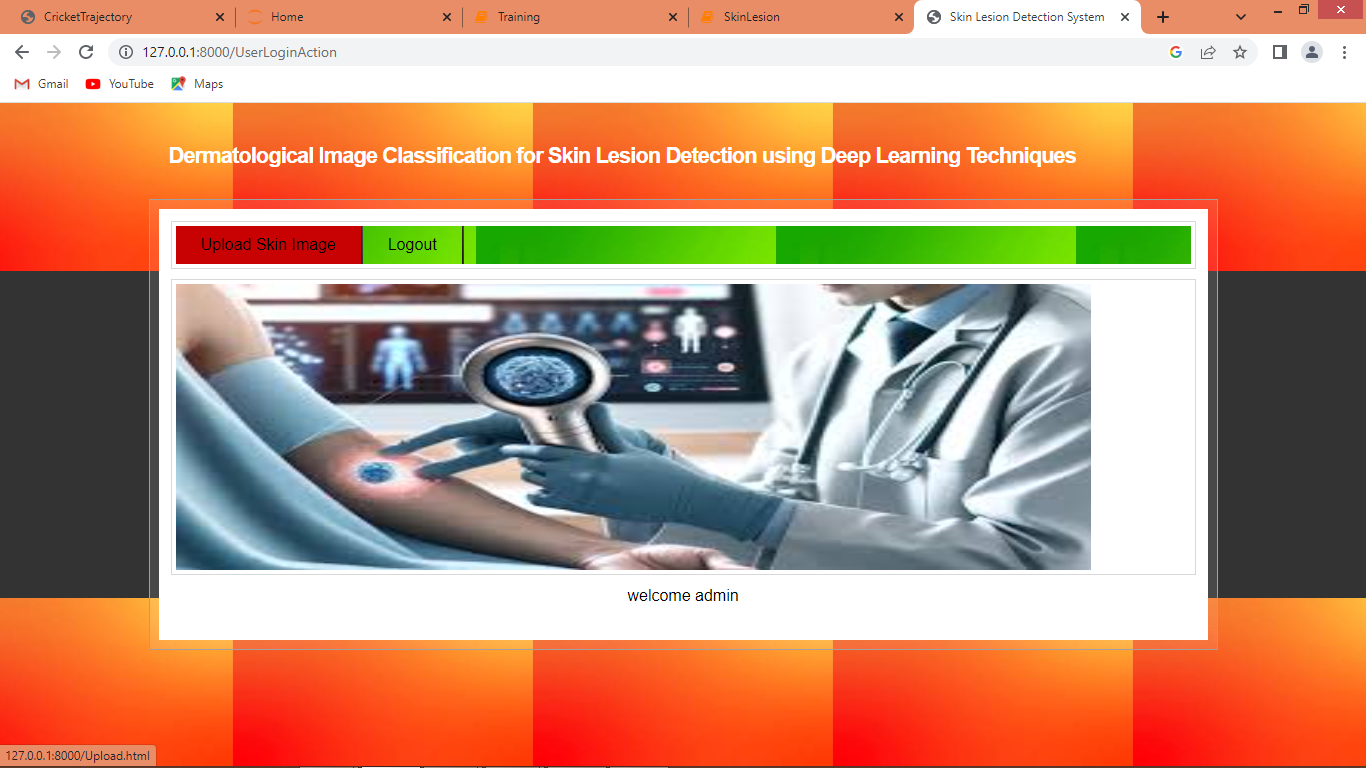
In above screen web server started and now open browser and then enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below page



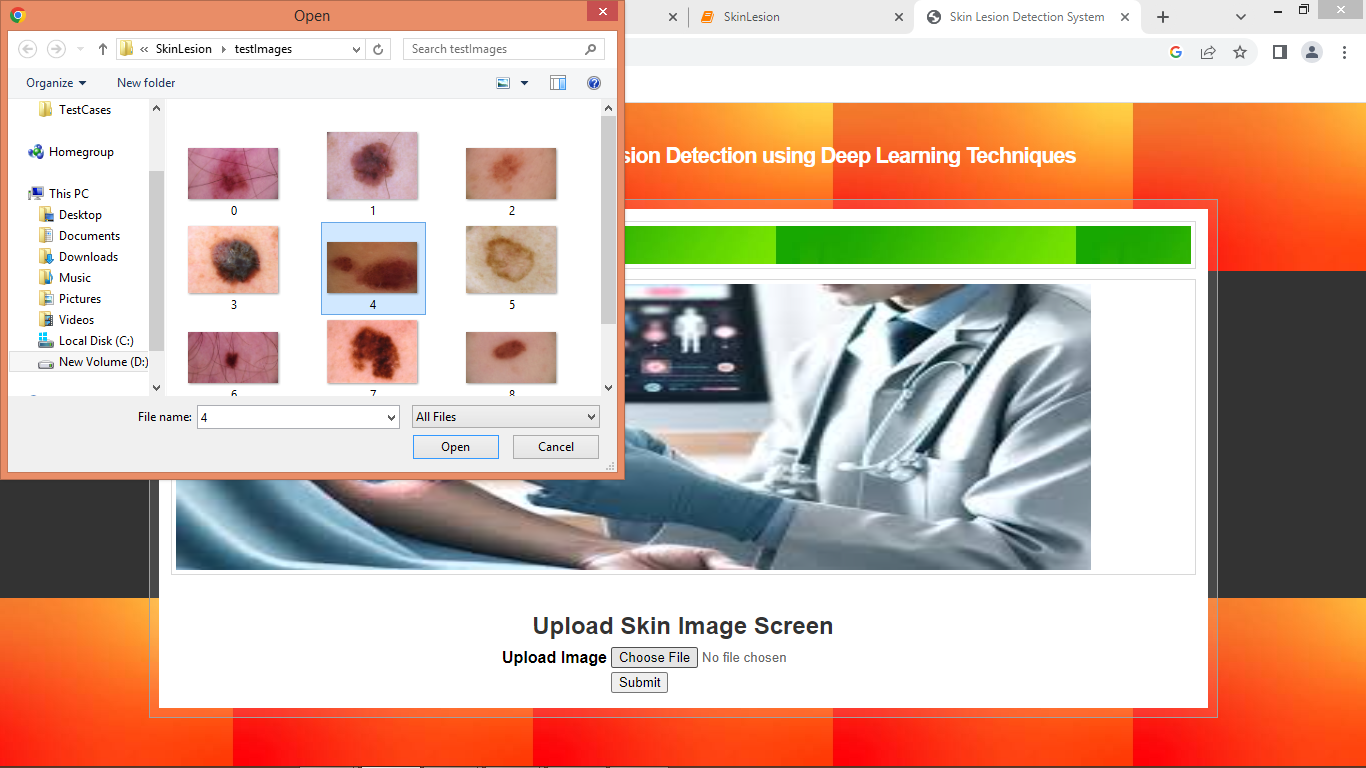
In above screen click on ‘User Login’ link to get below page



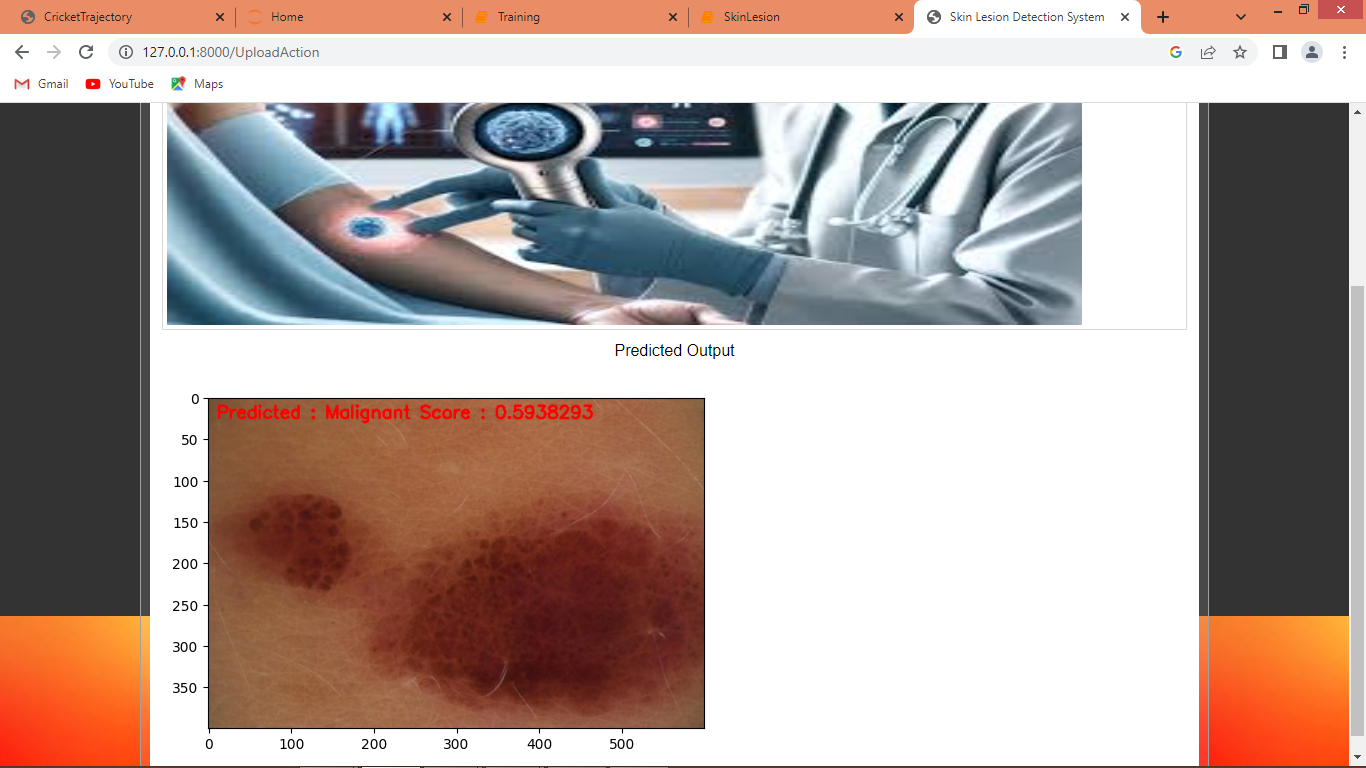
In above screen user can login by entering username and password as ‘admin and admin’ and after login will get below page



In above screen click on ‘Upload Skin Image’ link to get below page



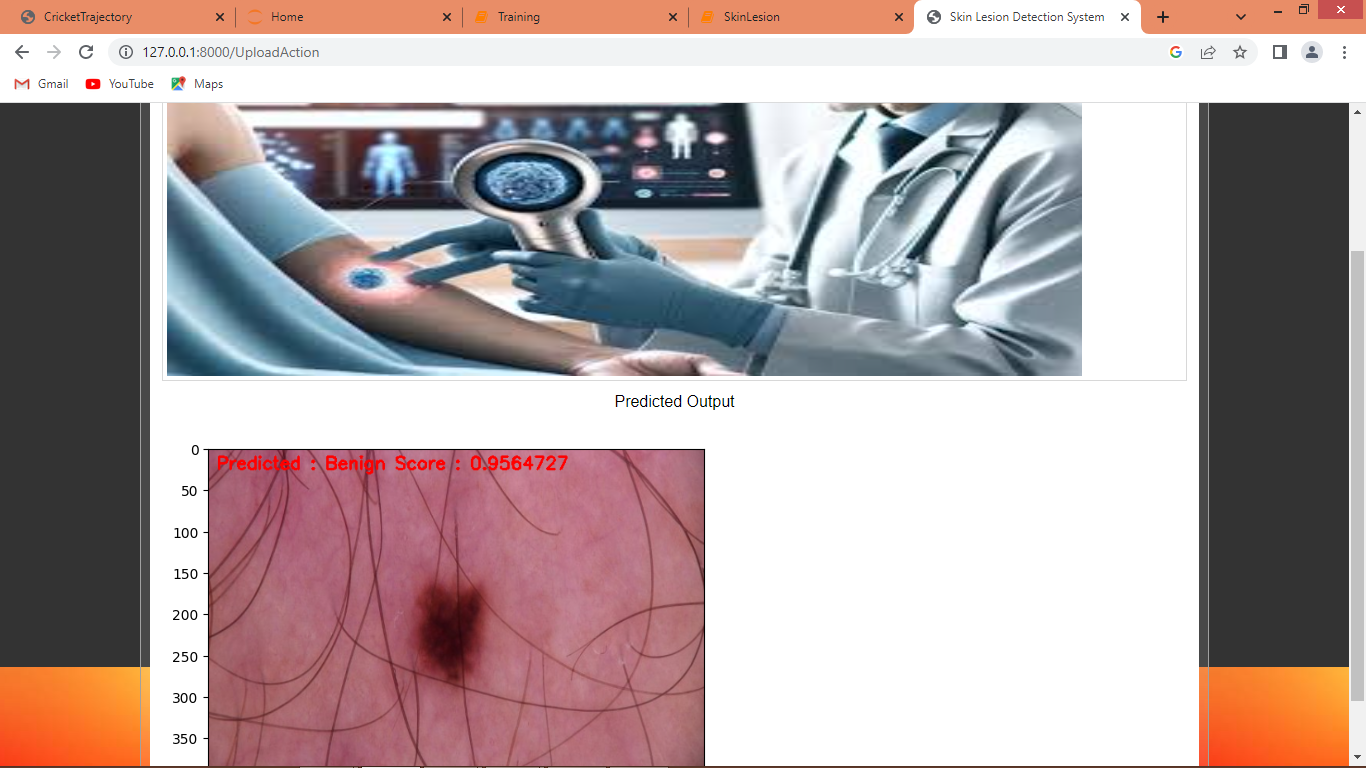
In above screen select and upload test image and then click on ‘Open and Submit’ button to get below page



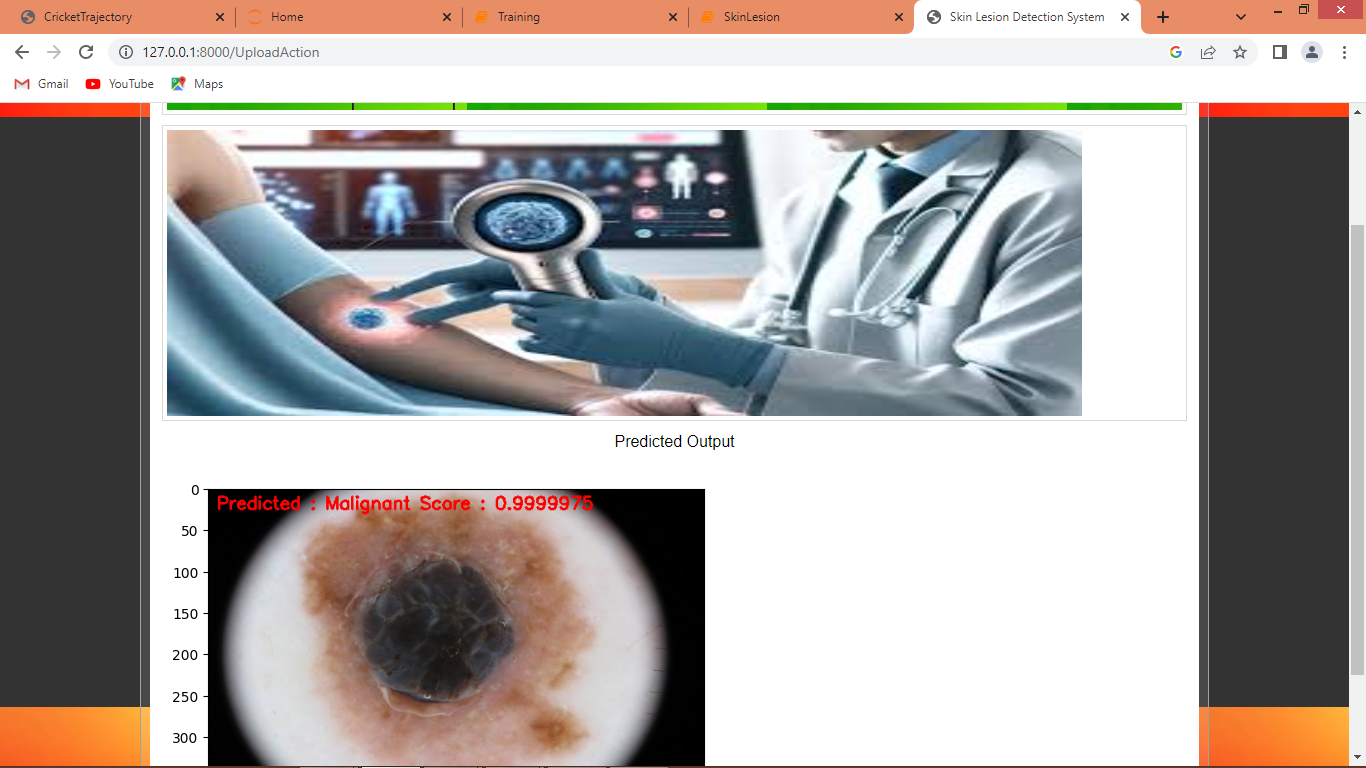
In above screen lesion predicted as Malignant



In above screen uploading another image



Above image predicted as Benign



Above image predicted as Malignant.

Similarly you can upload and test other images from JUPYTER or WEB