EyeDeep-Net: a multi-class diagnosis of retinal diseases using deep neural network

In propose paper author introducing deep neural network based algorithm to detect retinal diseases. To train network author employing RFMID dataset which contains retinal images of 46 different classes and from this classes author extracting most 4 different classes such as Normal, DR, ODC and MH. Above classes contains highly imbalance data where one class contains more number of images and other class contains fewer number of images and such data imbalance may affect accuracy so to enhance accuracy of propose algorithm author employing Data Augmentation techniques which will generate new synthetic images using available images. To generate augmented images author has used left and right rotation of 15 % with sheer rotation as 0.8 and applied horizontal flip.

Retinal disease are one of the dangerous disease which may cause blindness if not detected and treat early. So to enhance detection accuracy author employed EyeDeep-Net model. The objective of EyeDeep-Net model is to develop a diagnostic framework for the diagnosis of multiple fundus disease at an early stage through a common deep neural network so that people can get treatment on time and take necessary actions to save their eyes from being lost. The main contribution of propose algorithm is to developed robust and effective framework using the proposed EyeDeep-Net model, a deep learning architecture to classify fundus images and diagnose different eye diseases. An open source multi-labelled dataset was transformed into a multiclass dataset and then extracted fundus images were augmented to deal with real-world conditions and processed.

The proposed EyeDeep-Net architecture comprises various 2-dimensional convolutional layers, max-pooling layers, and batch normalisation layers which were later fine-tuned with selective hyper-parameters for accurate feature extraction from the input fundus images in different categories. A fully connected layer was added as a classifier to diagnose the eye disease that takes extracted feature maps from CNN as an input for classification.

Extension Concept:

In propose EyeDeep-Net model author has used padding as ‘SAME’ which adds additional rows and columns of pixels around the edges of the input data so that the size of the output feature map is the same as the size of the input data. Adding additional rows and columns may hinder algorithm performance so we can apply padding as ‘VALID’ which is used when it is desired to reduce the size of the output feature map in order to reduce the number of parameters in the model and improve its computational efficiency. When features size reduced then algorithm will get more optimized features and accuracy will get enhanced.

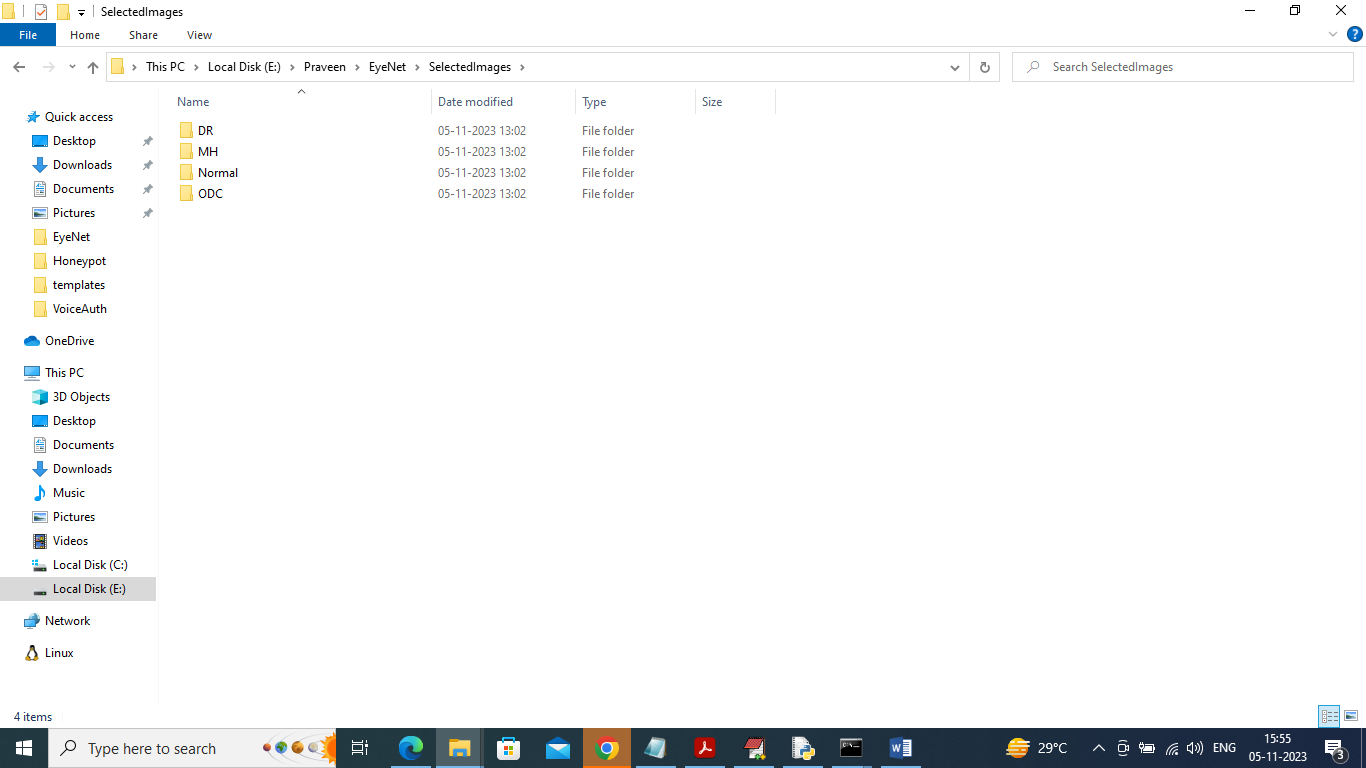
So extension work we have utilized padding same with valid to reduce parameters and to enhance accuracy.

To train propose and extension model we have utilized same dataset send by your which is downloaded from below URL

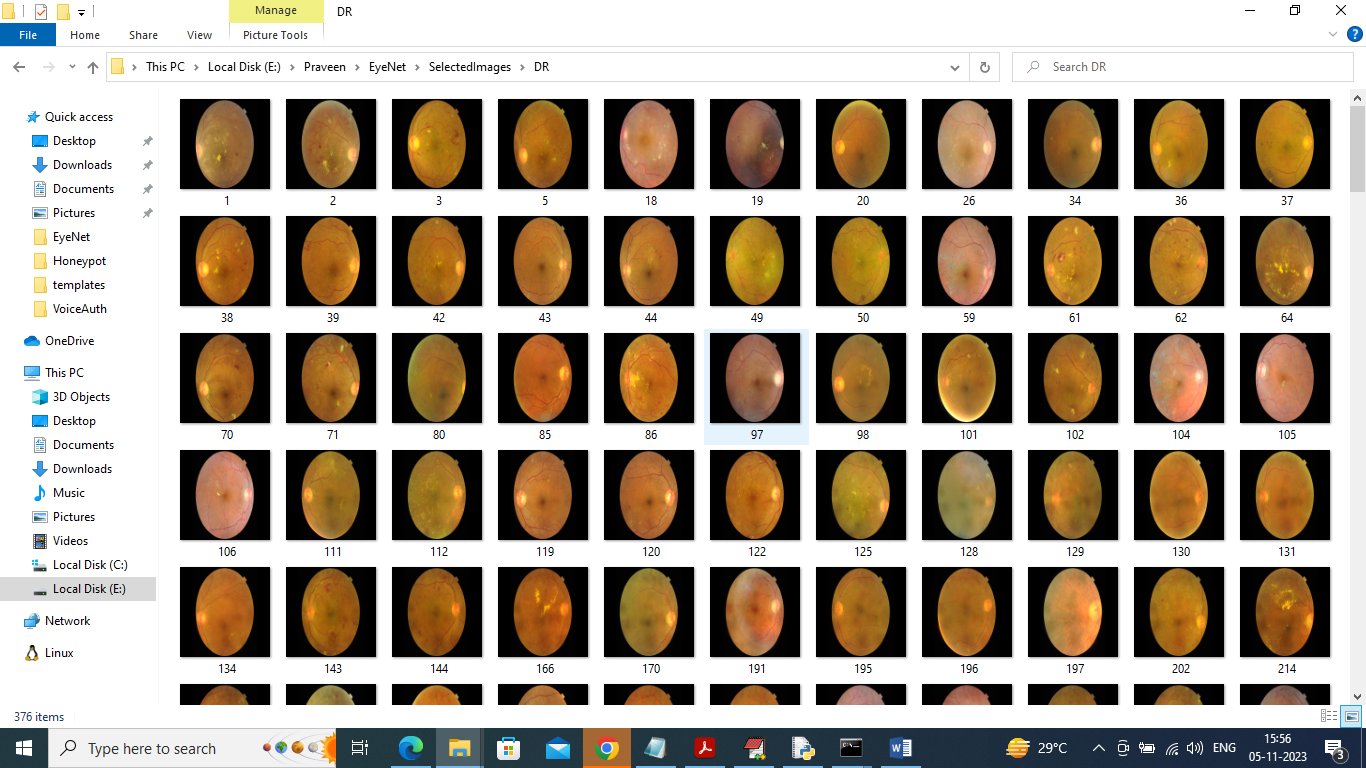
<https://riadd.grand-challenge.org/download-all-classes>

From this dataset we have extracted 4 different classes and then train with propose and extension model. Propose algorithm author has trained with ADAM and SGD optimizer and then evaluate its performance in terms of accuracy, precision, recall, confusion matrix and FSCORE.

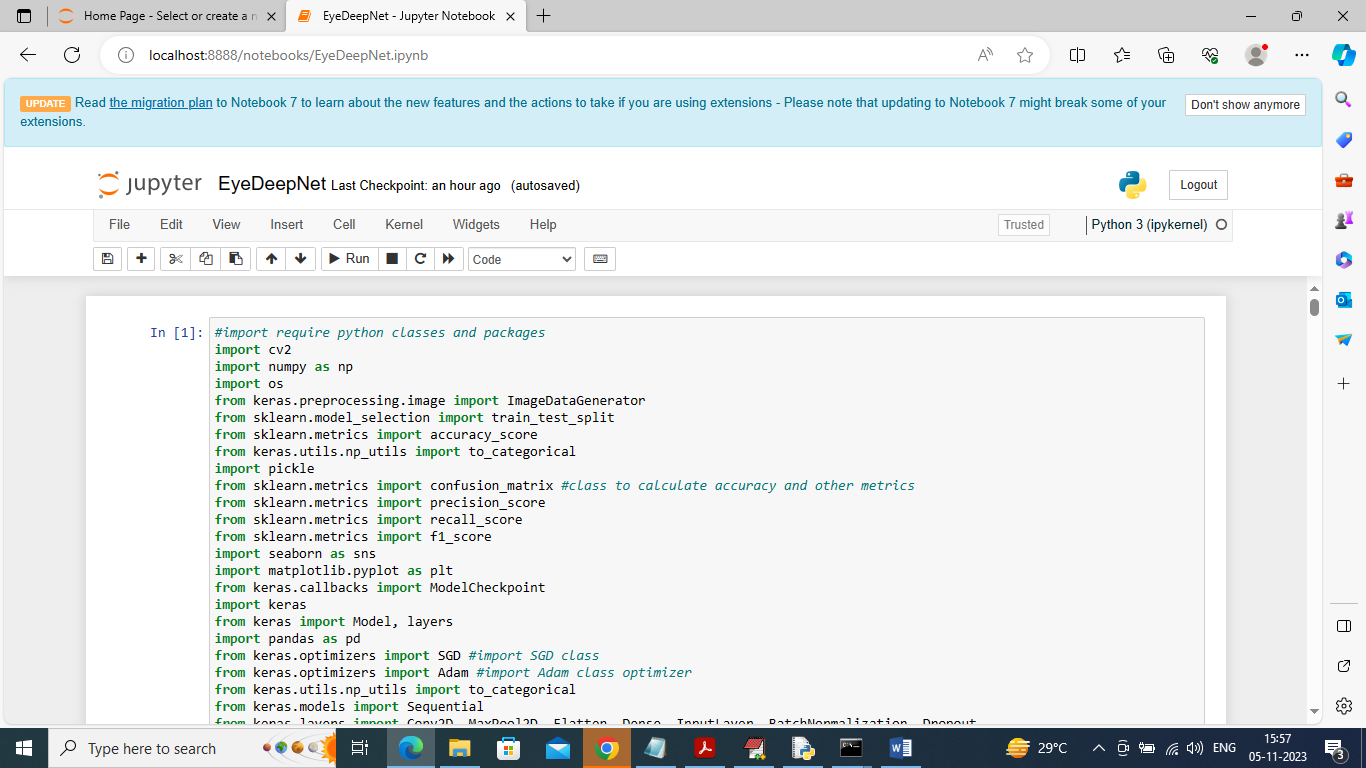
In below screen we are showing dataset details which extracted from given dataset



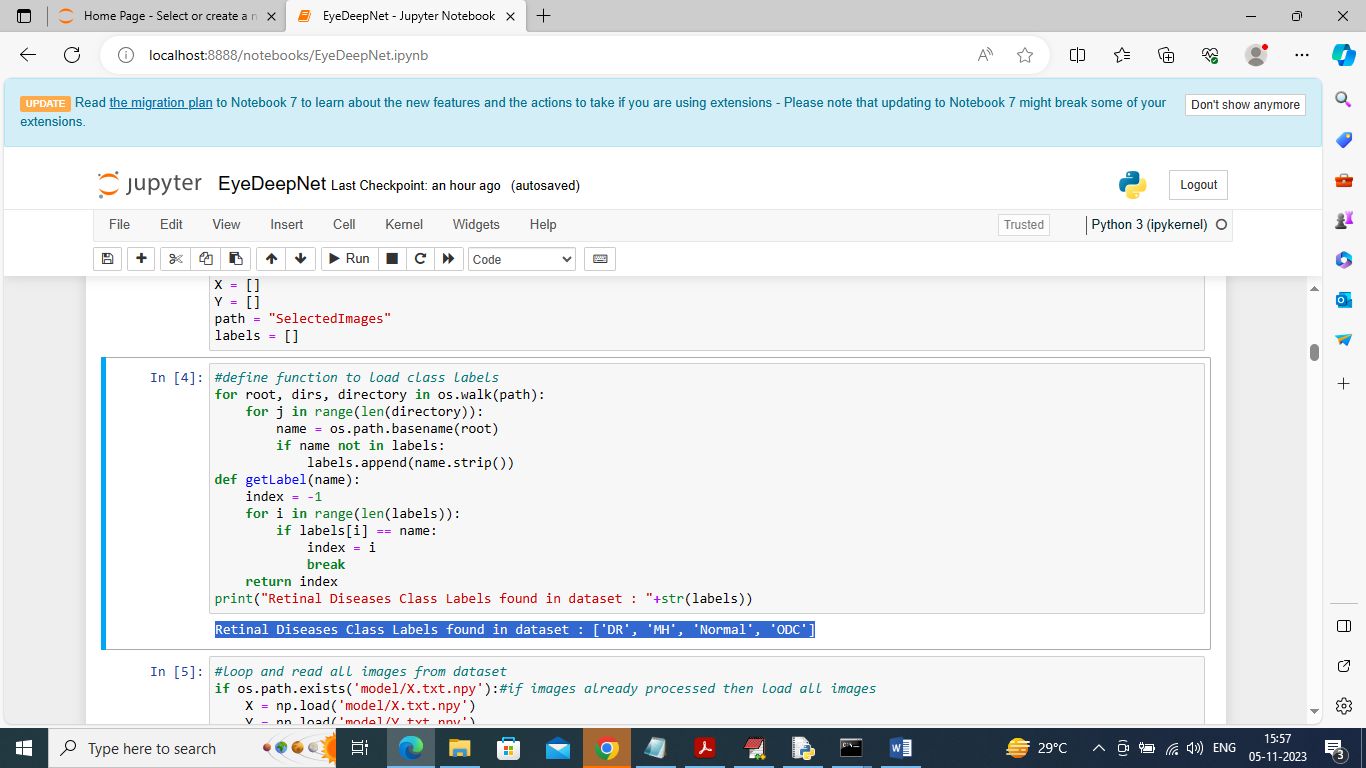
In above screen we have 4 different folders and just go inside any folder to view related images



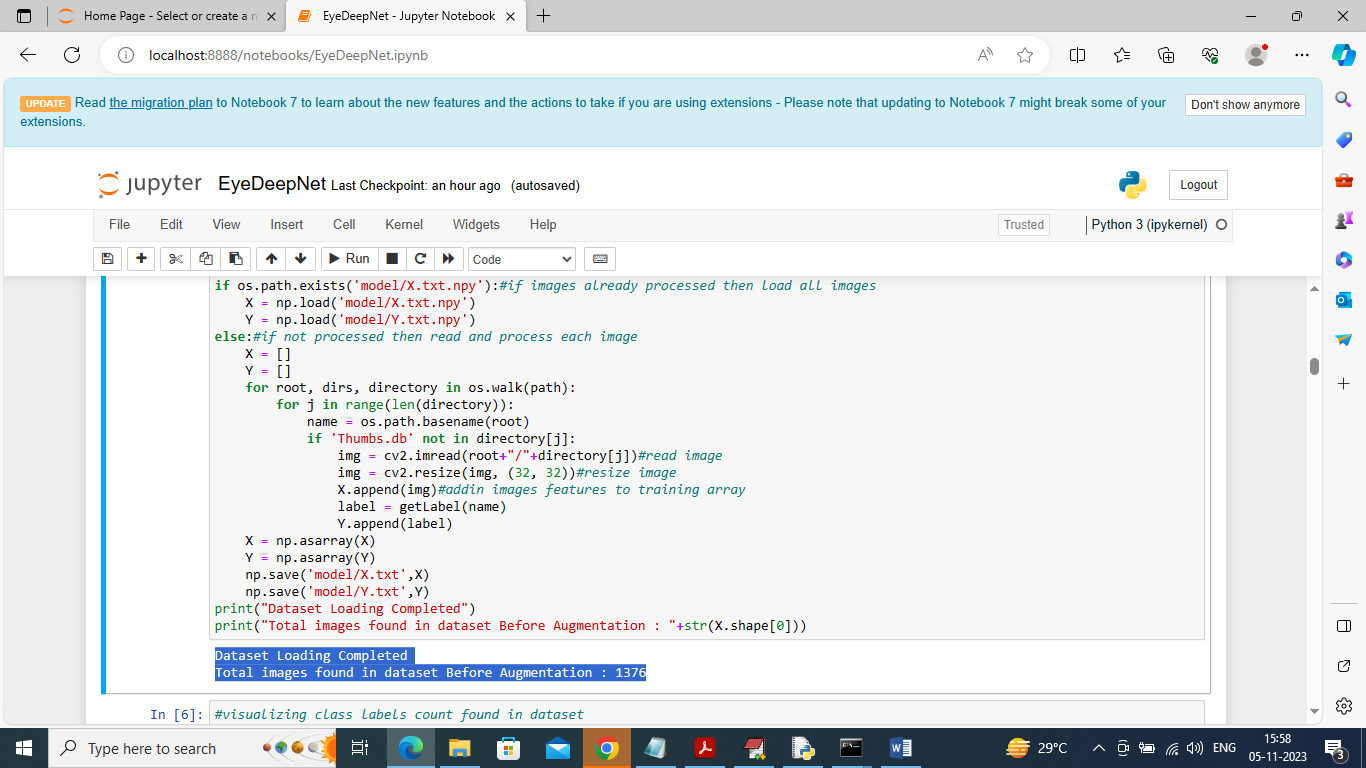
So by using above images we will train and test algorithm performance and we have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



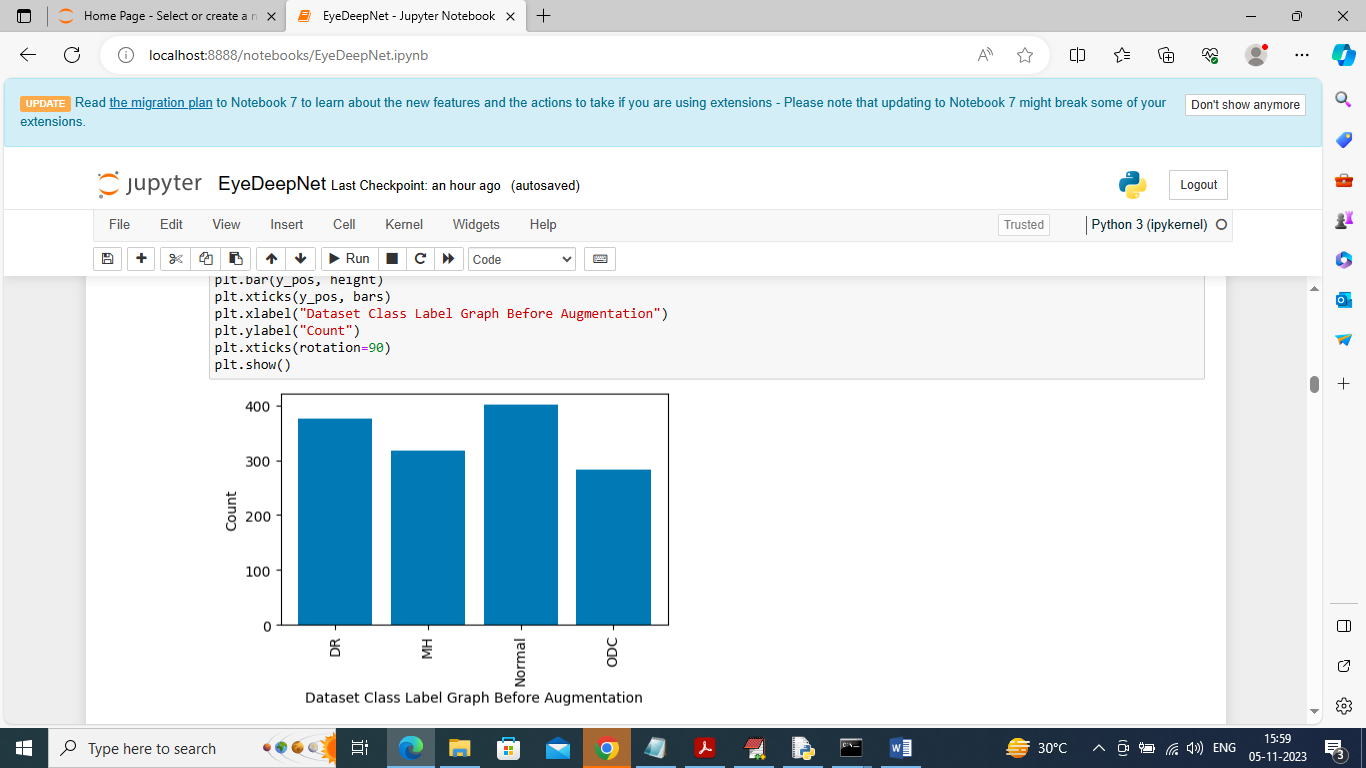
In above screen loading required python classes and packages



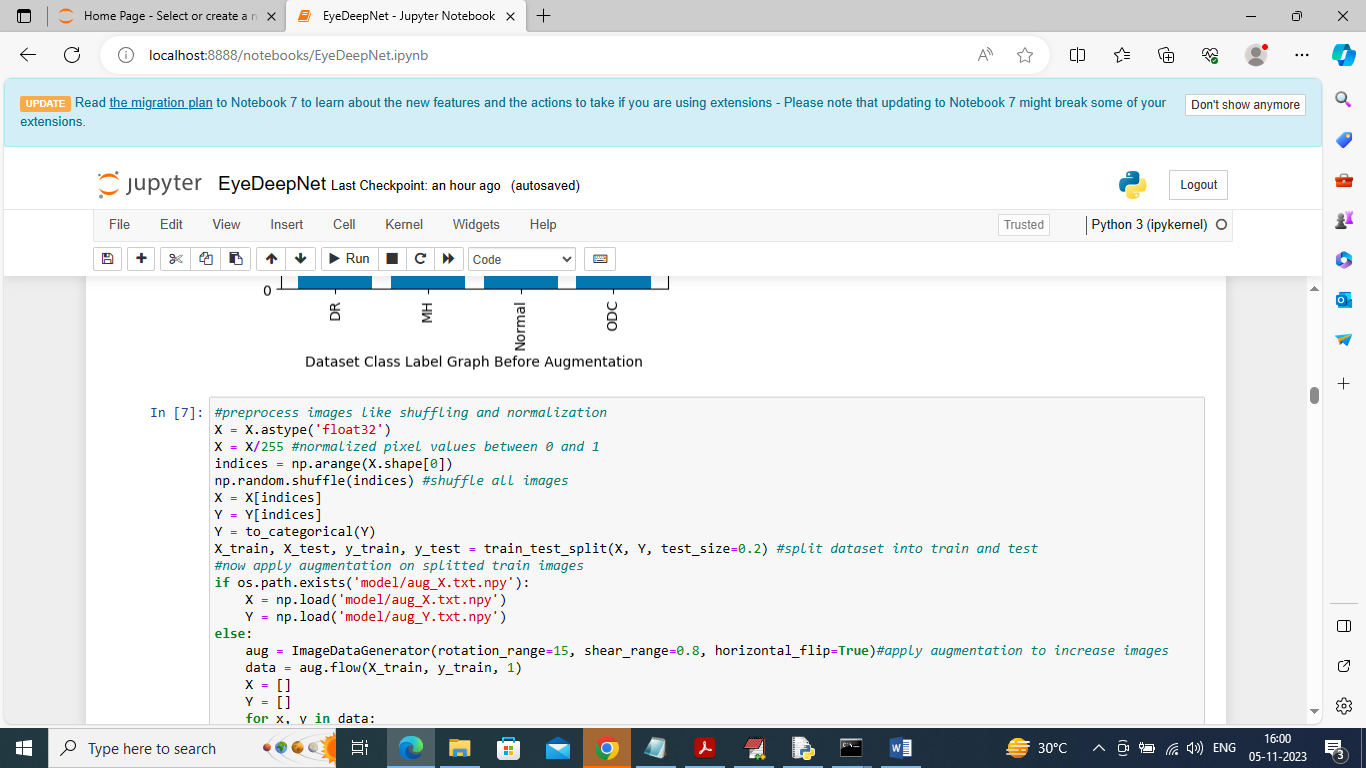
In above screen in blue colour text displaying different class labels found in dataset



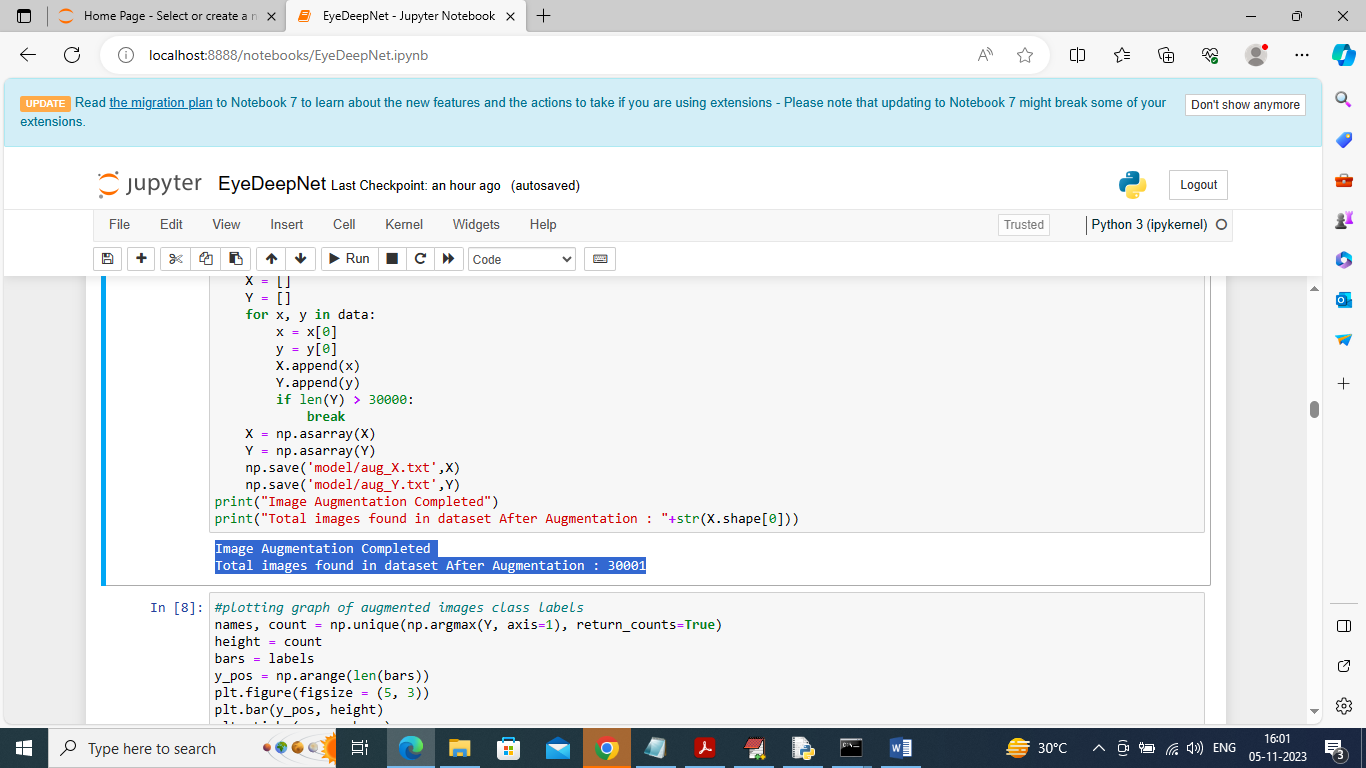
In above screen reading all images from dataset and then resizing and adding to training array and after loading we can see dataset contains 1376 images without augmentation and this output displaying in blue colour text



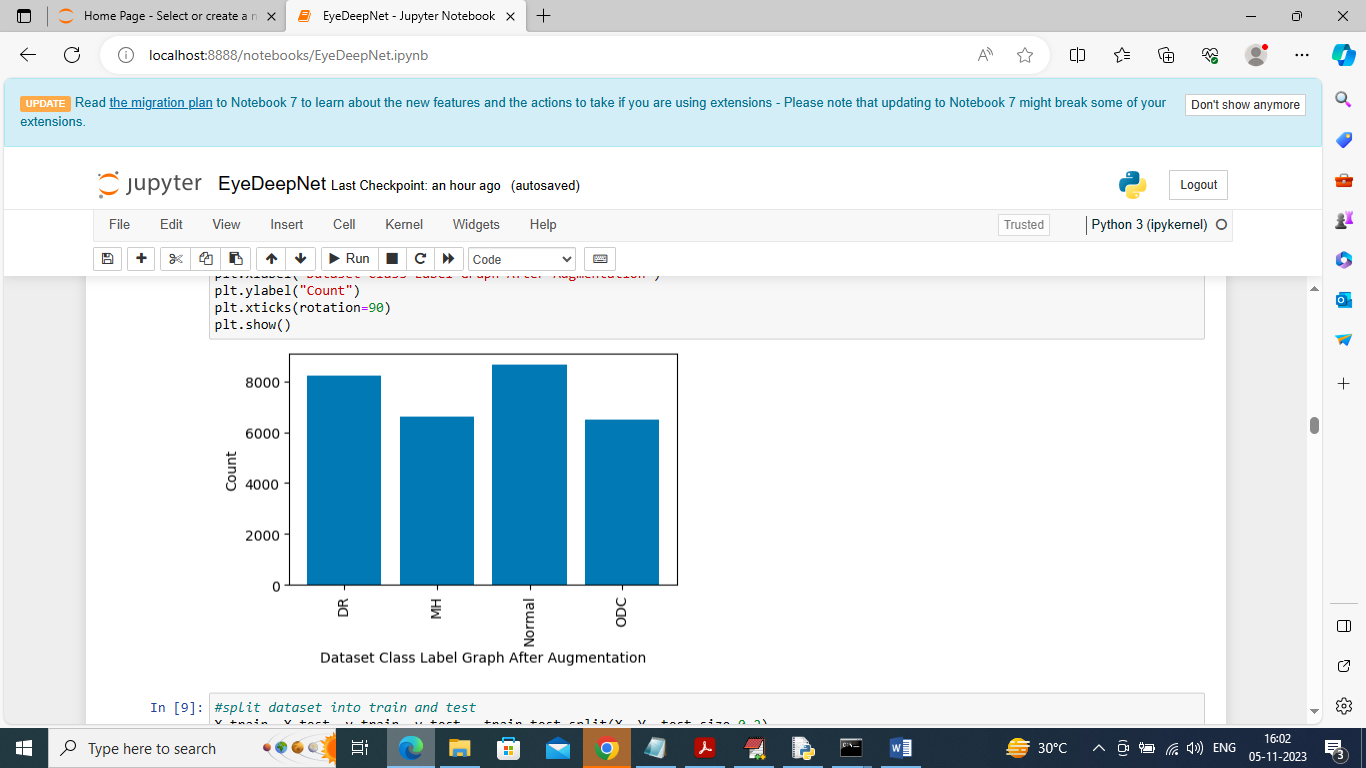
In above screen displaying number of images in dataset in graph format in different classes where x-axis represents class label and y-axis represents number of images in that class



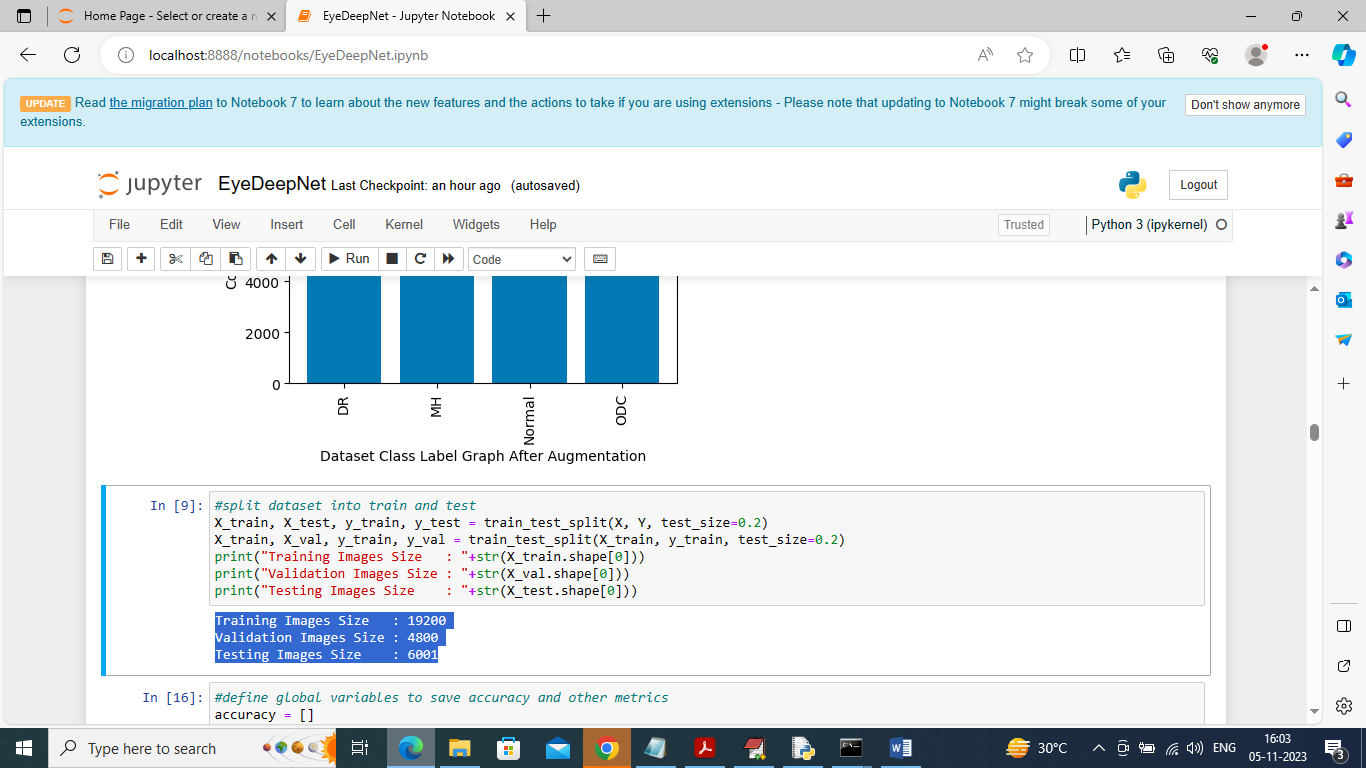
In above code we are processing dataset such as normalization, shuffling, splitting to train and test and then applying augmentation technique to increase dataset size and after executing this block will get below output



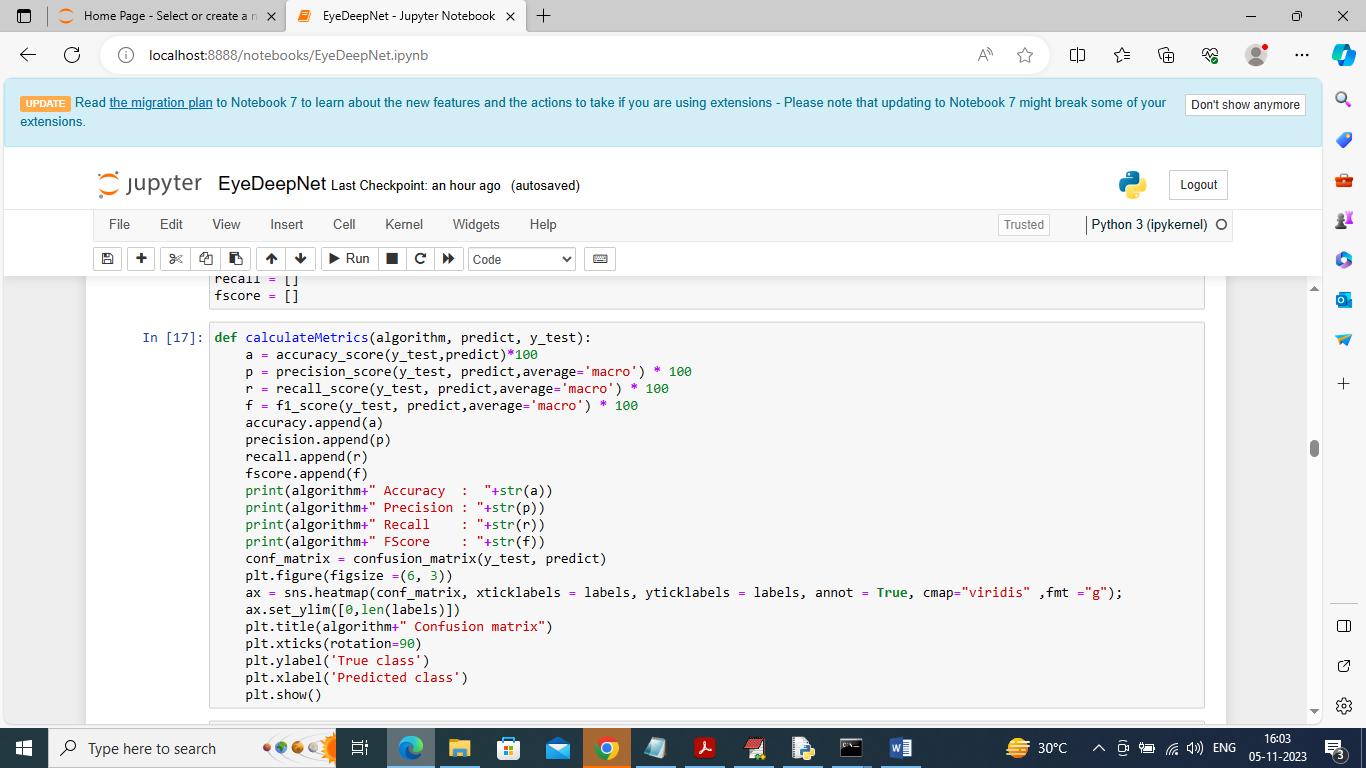
In above screen in blue colour text can see after augmentation dataset images size increased to 30001 and below is the graph



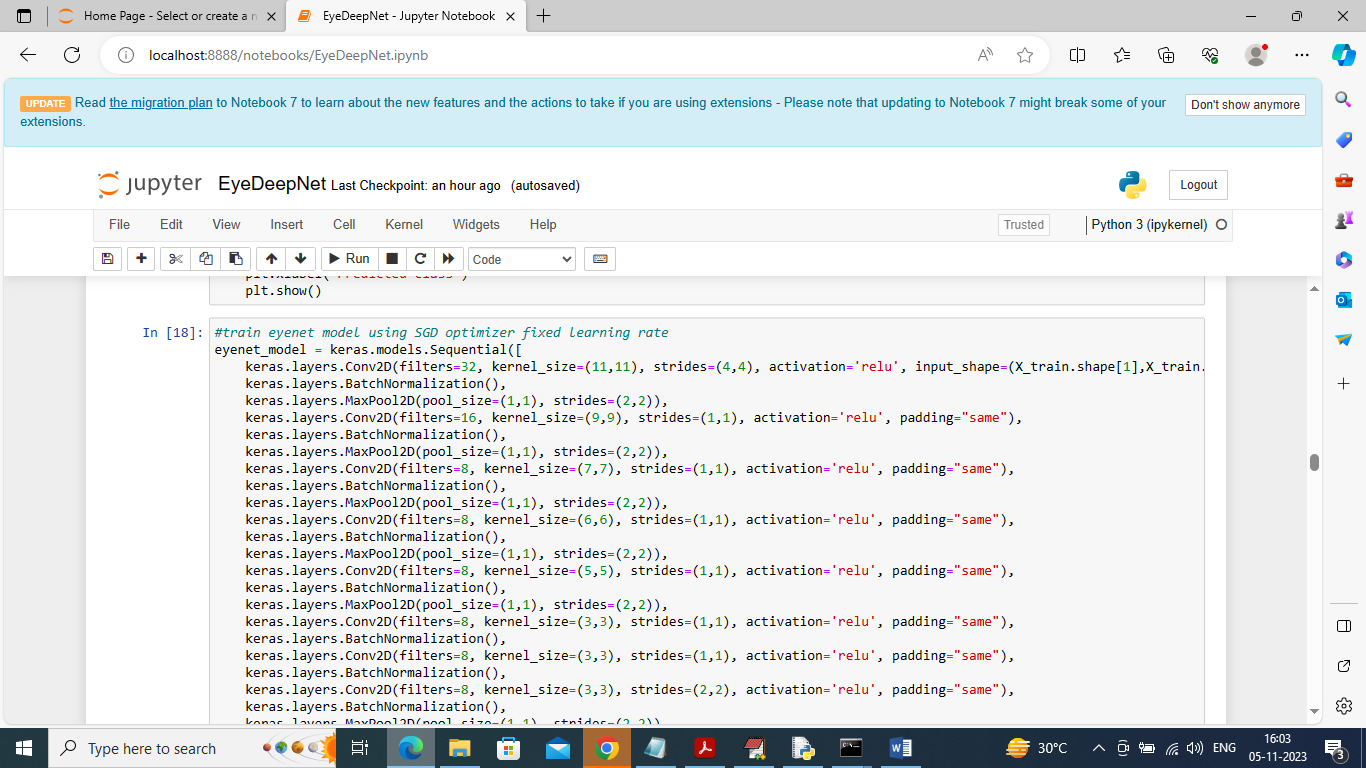
In above screen after augmentation we can see number of samples in each class



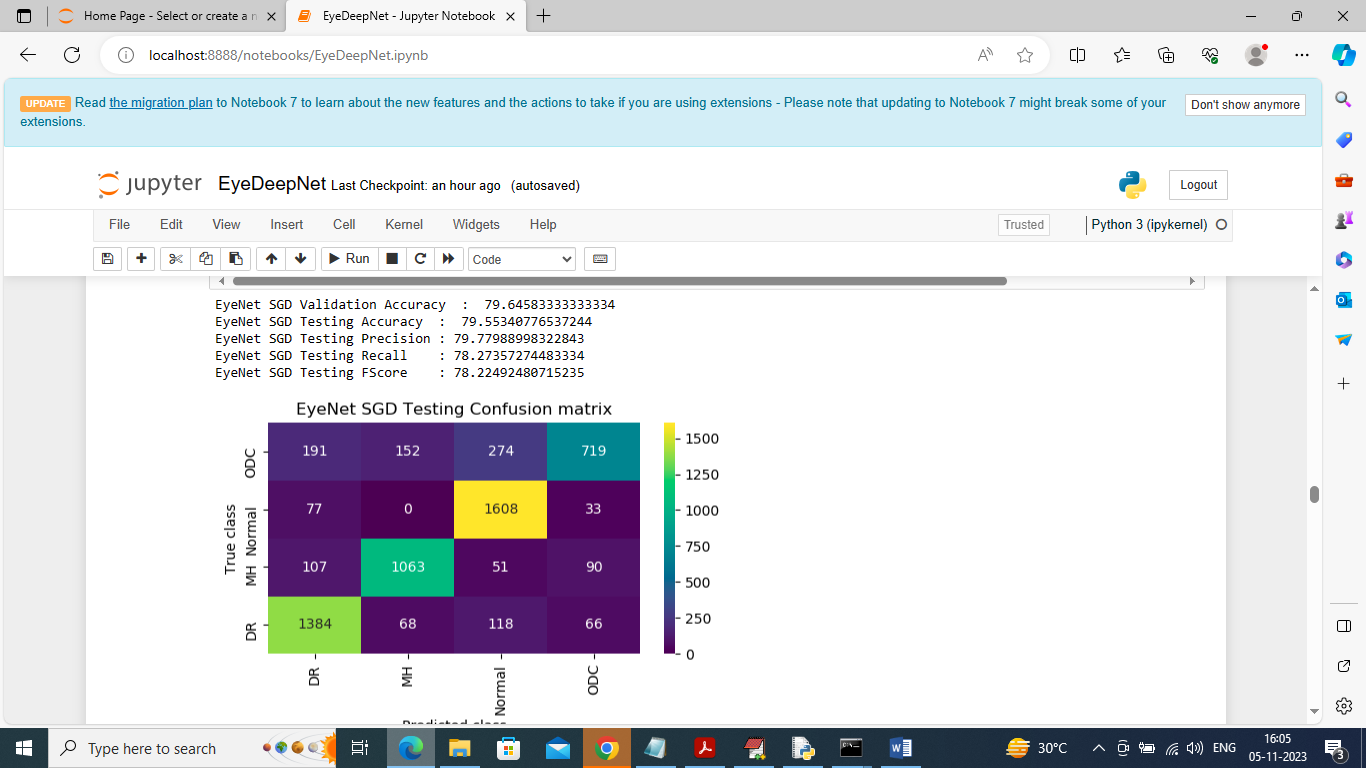
In above screen splitting dataset into train, test and validation part and in blue colour we can see their size



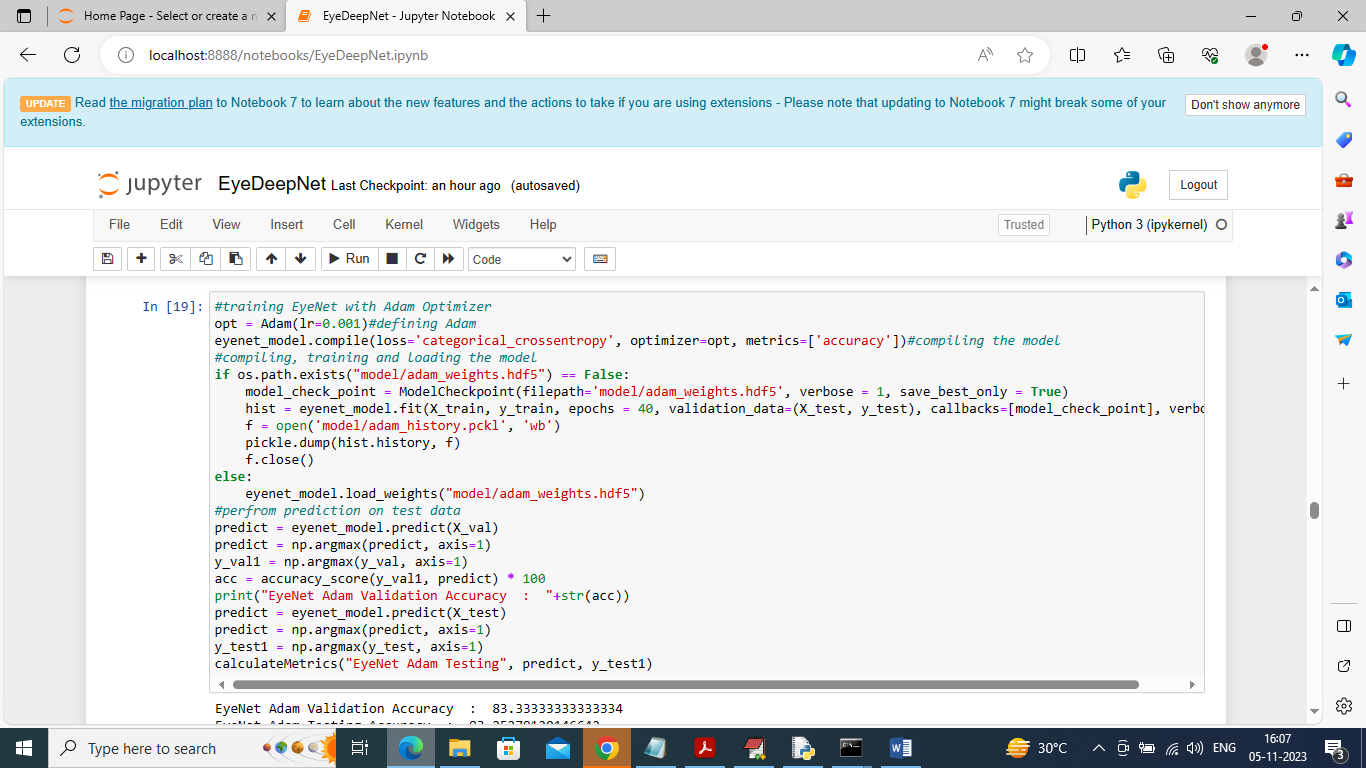
In above screen defining function to calculate accuracy and other metrics



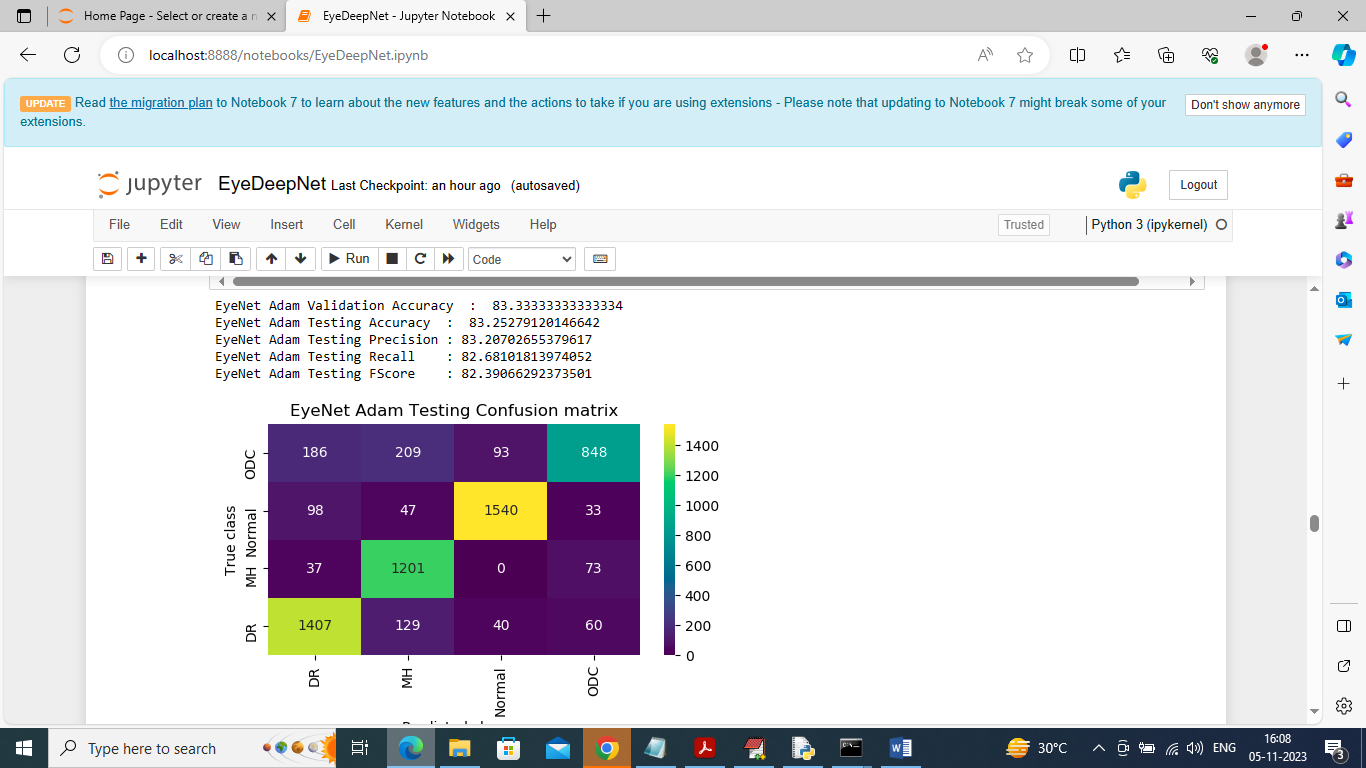
In above screen defining EYENET model with RELU, padding, strides and SGD optimizers and after executing this model will get below output



In above screen EYENET with SGD got 79.64% as validation accuracy and 79.55% as the testing accuracy and can see other metrics also. In confusion matrix graph x-axis represents True Labels and y-axis represents Predicted Labels and different colour boxes in diagnol represents correct prediction count and remaining blue boxes contains incorrect prediction count which are very few



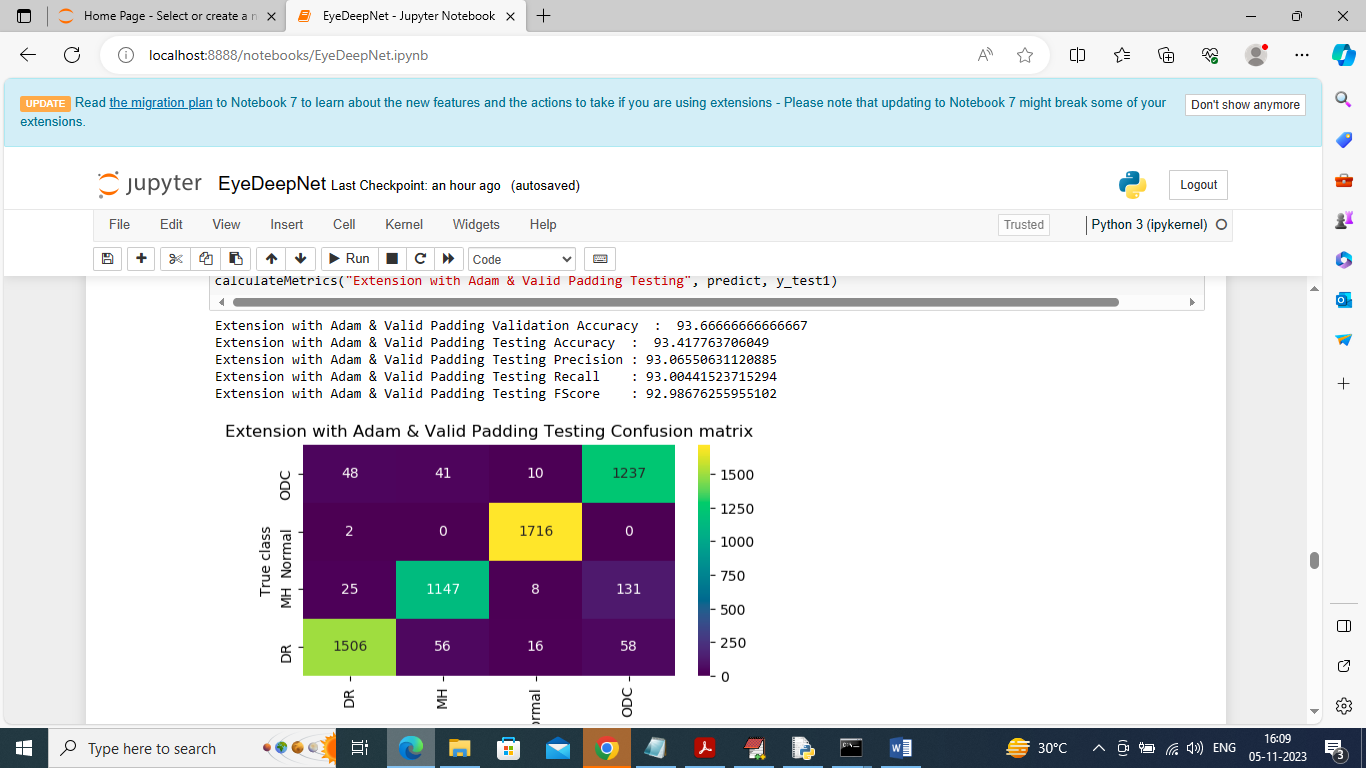
In above screen training EYENET with ADAM optimizer and after executing this block will get below output



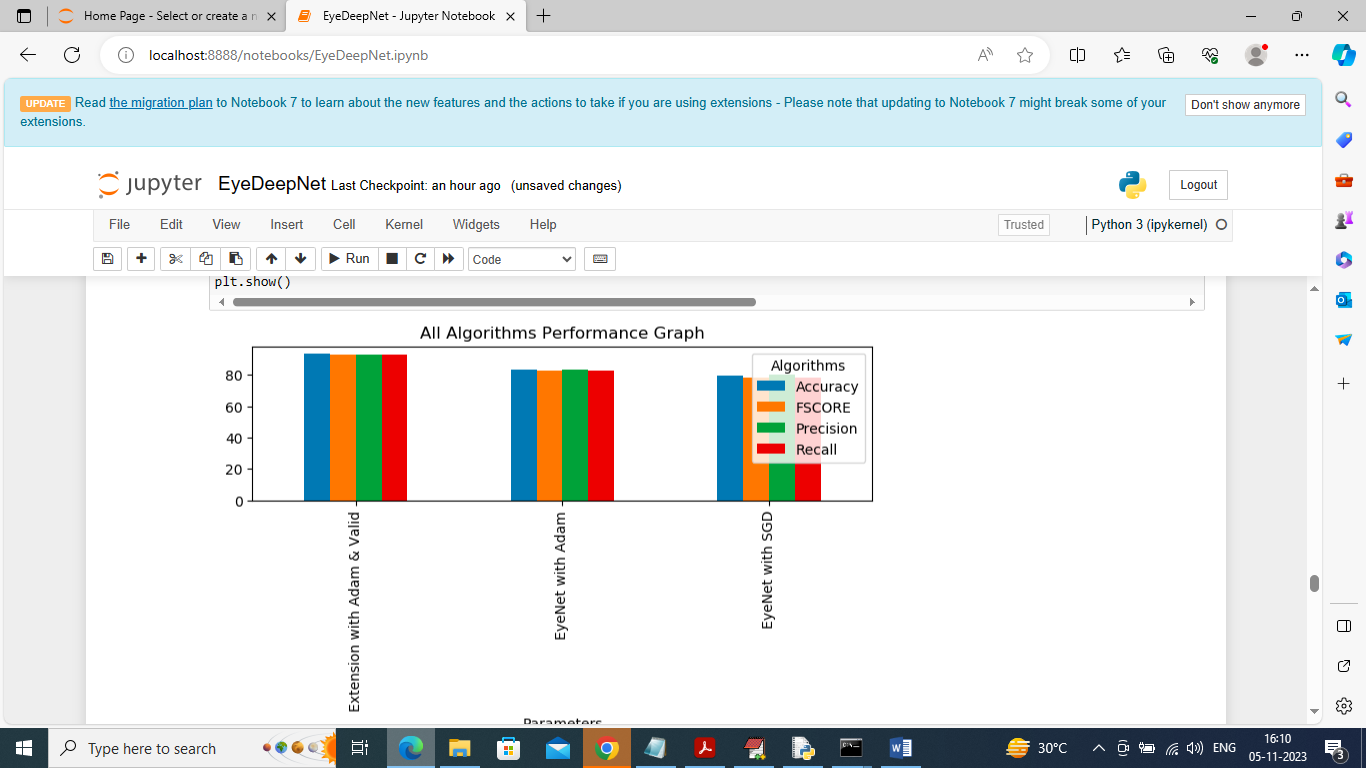
In above screen EYENET with ADAM got 83.33 as validation accuracy and 83.25 as the testing accuracy and can see other metrics and graph



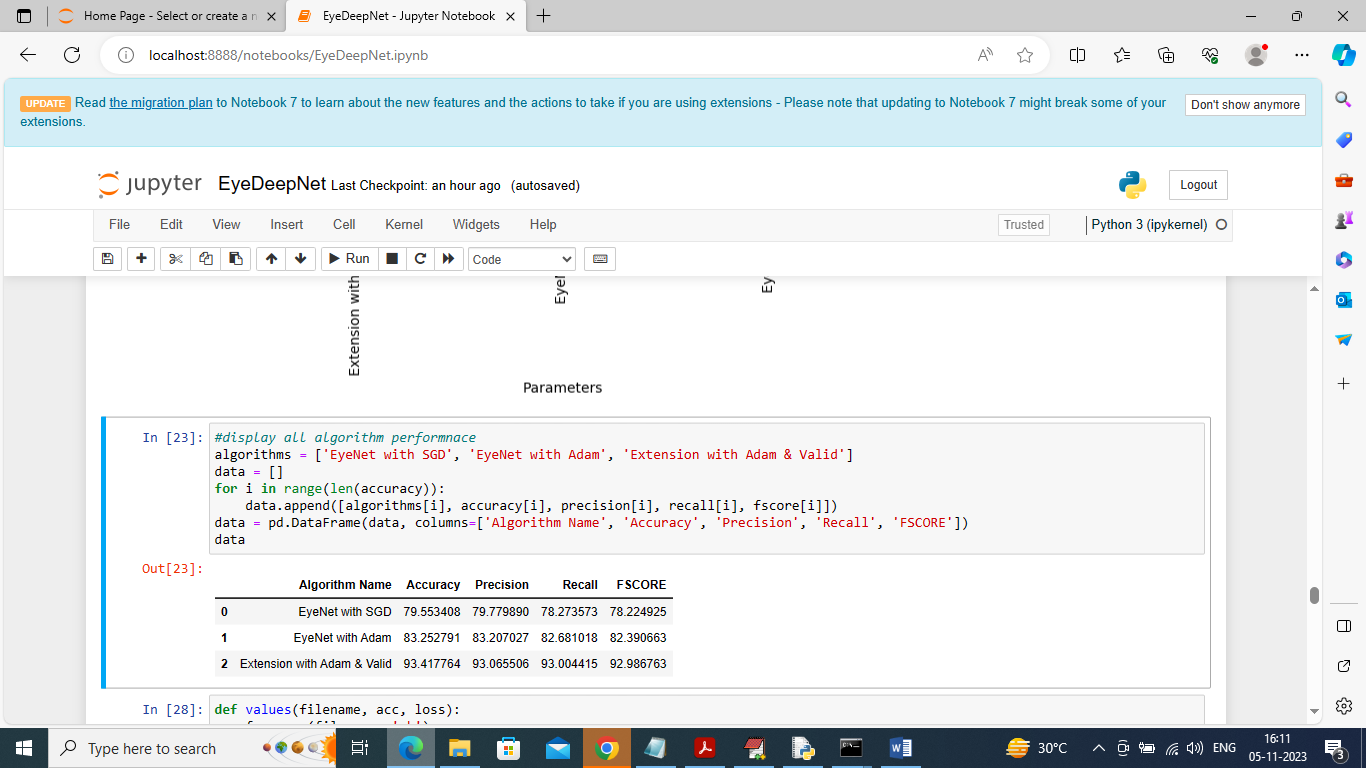
In above screen training extension model with padding as ‘same and valid’ and after executing this model will get below output



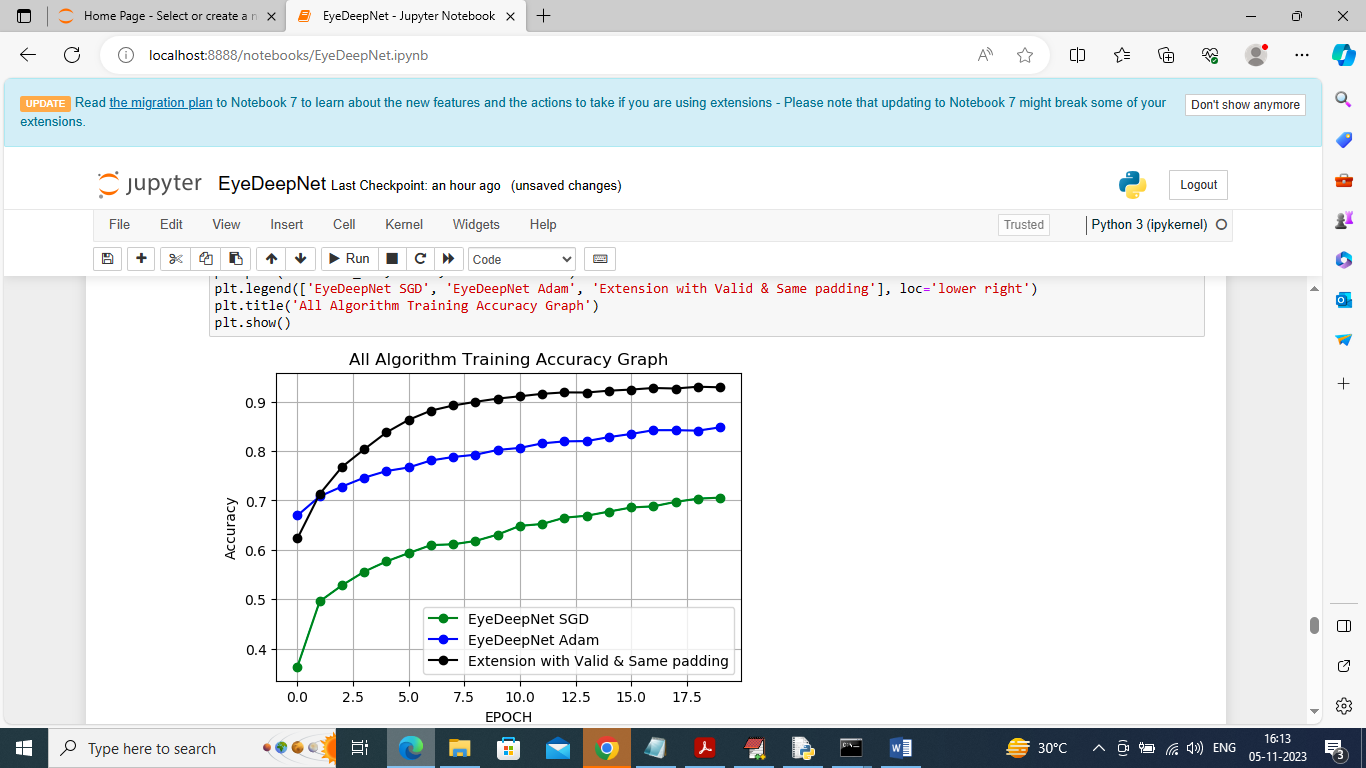
In above screen extension model got 93.66% as validation accuracy and 93.41 as the training accuracy and can see other metrics and confusion matrix graph



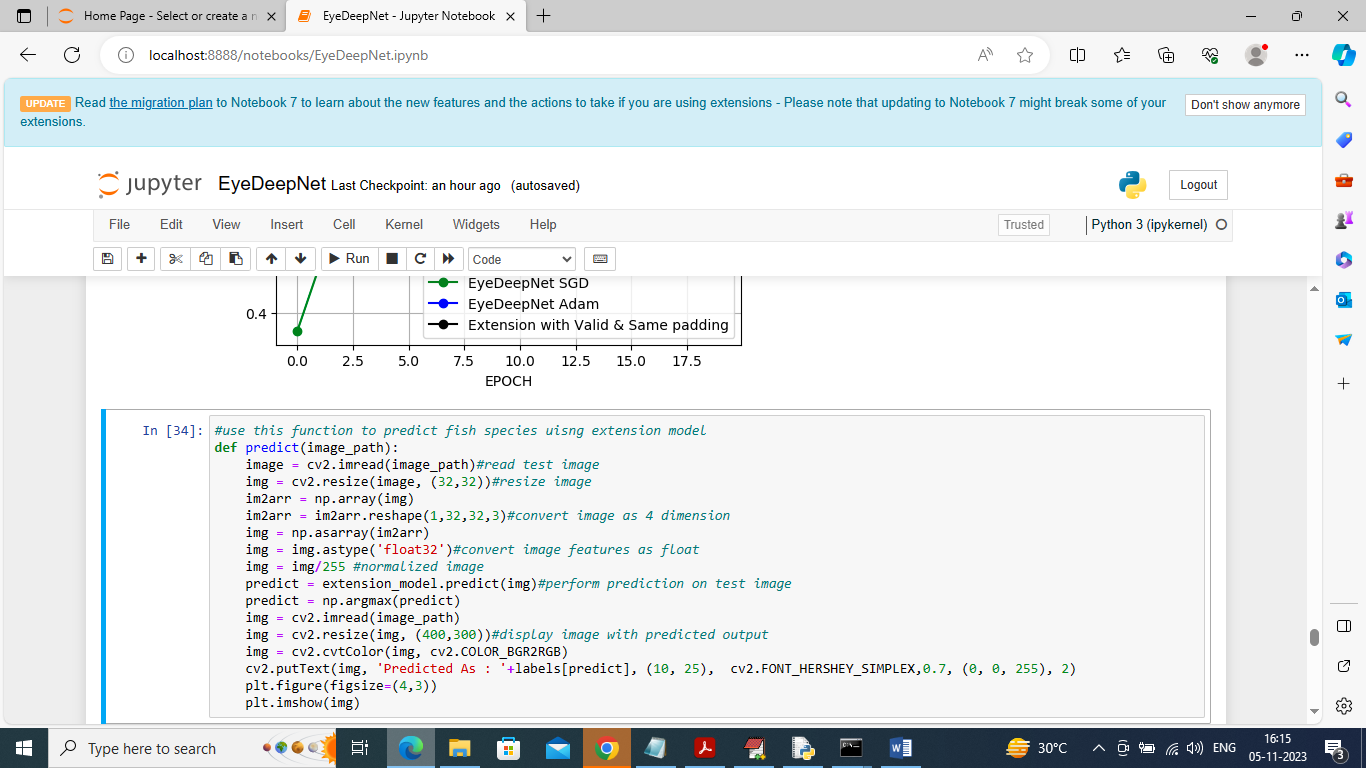
In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithm extension model got high accuracy



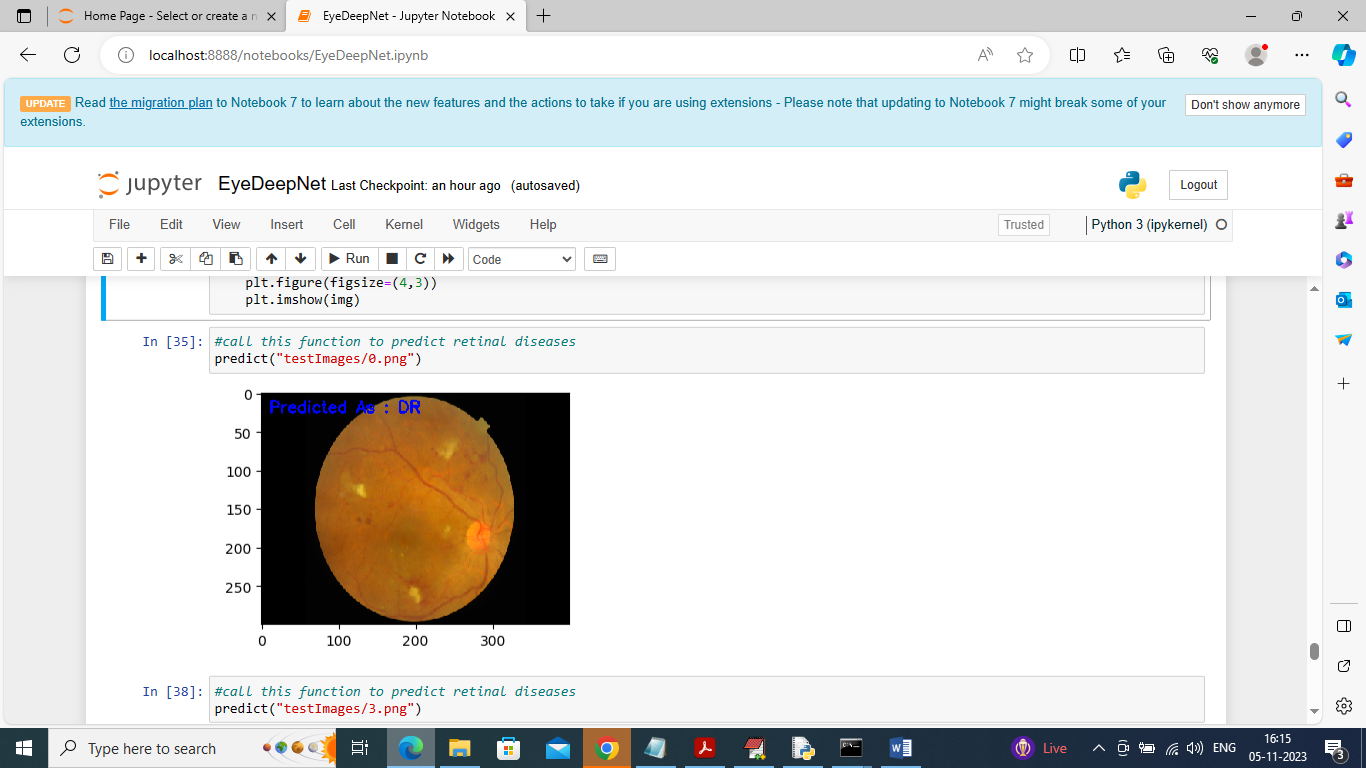
In above screen showing all algorithm performance in tabular format



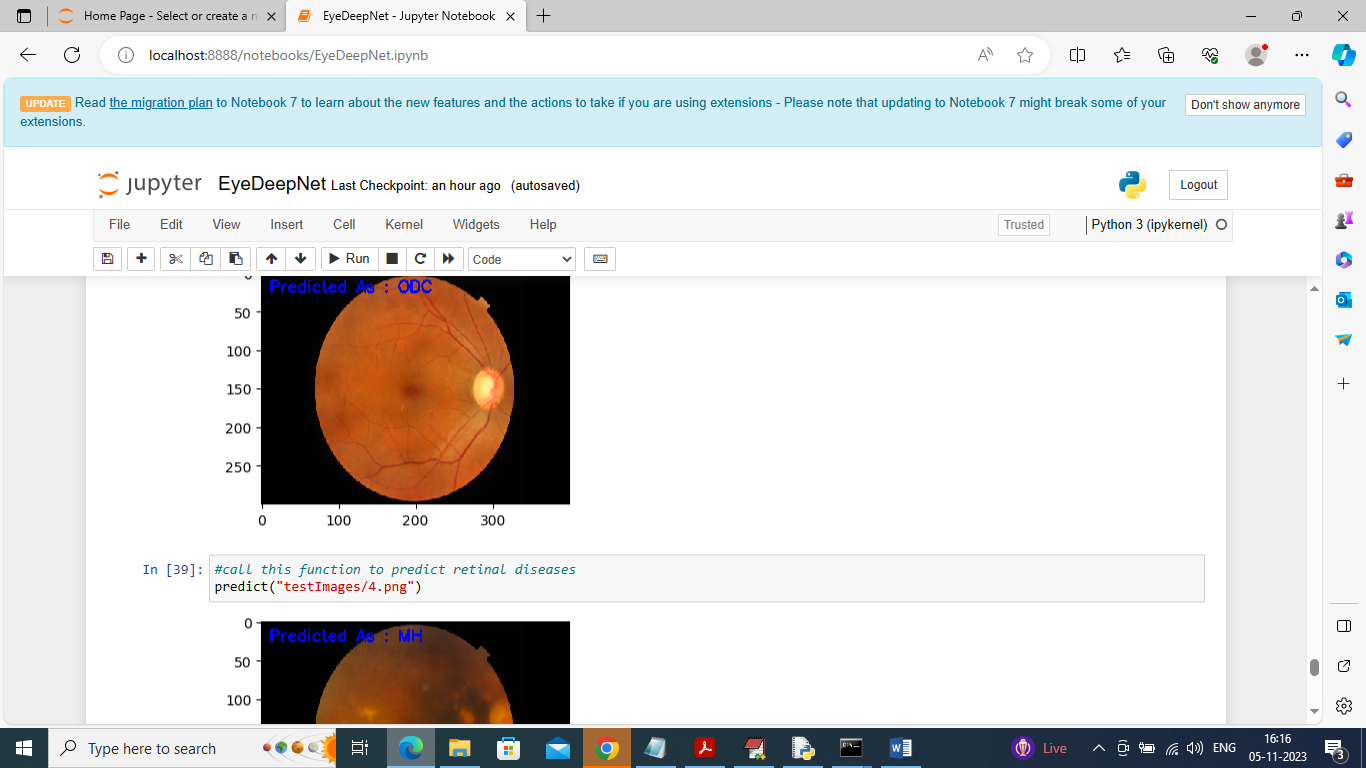
In above screen displaying training accuracy graph of propose EYENET with Adam (blue line) and SGD (green line) and extension model (black line). In above graph x-axis represents training Epoch and y-axis represents accuracy and in above graph with each increasing epoch accuracy got increase for all 3 algorithms

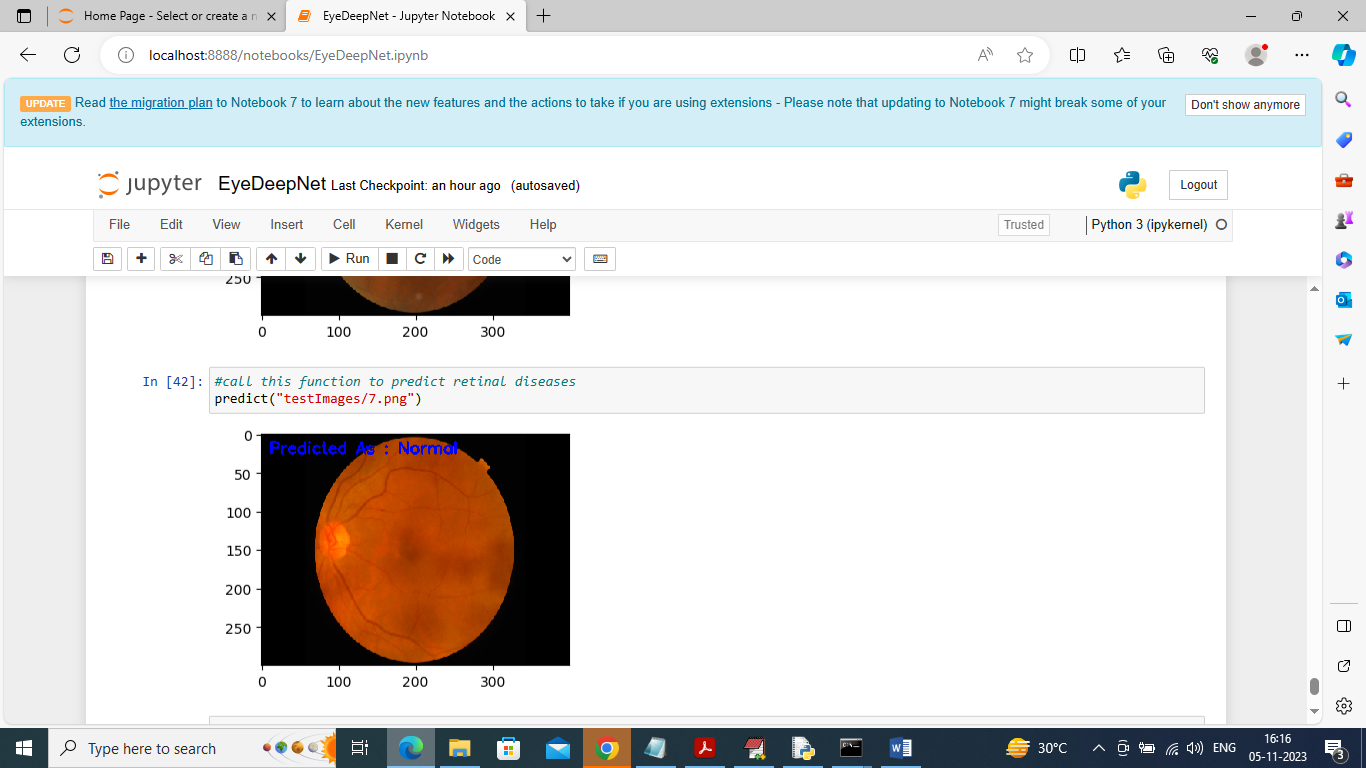


In above screen defining function to predict retina disease from test image



In above screen calling predict function with test image path and in blue colour text we can see predicted output as retinal disease





In above screen can see predicted output from other test images