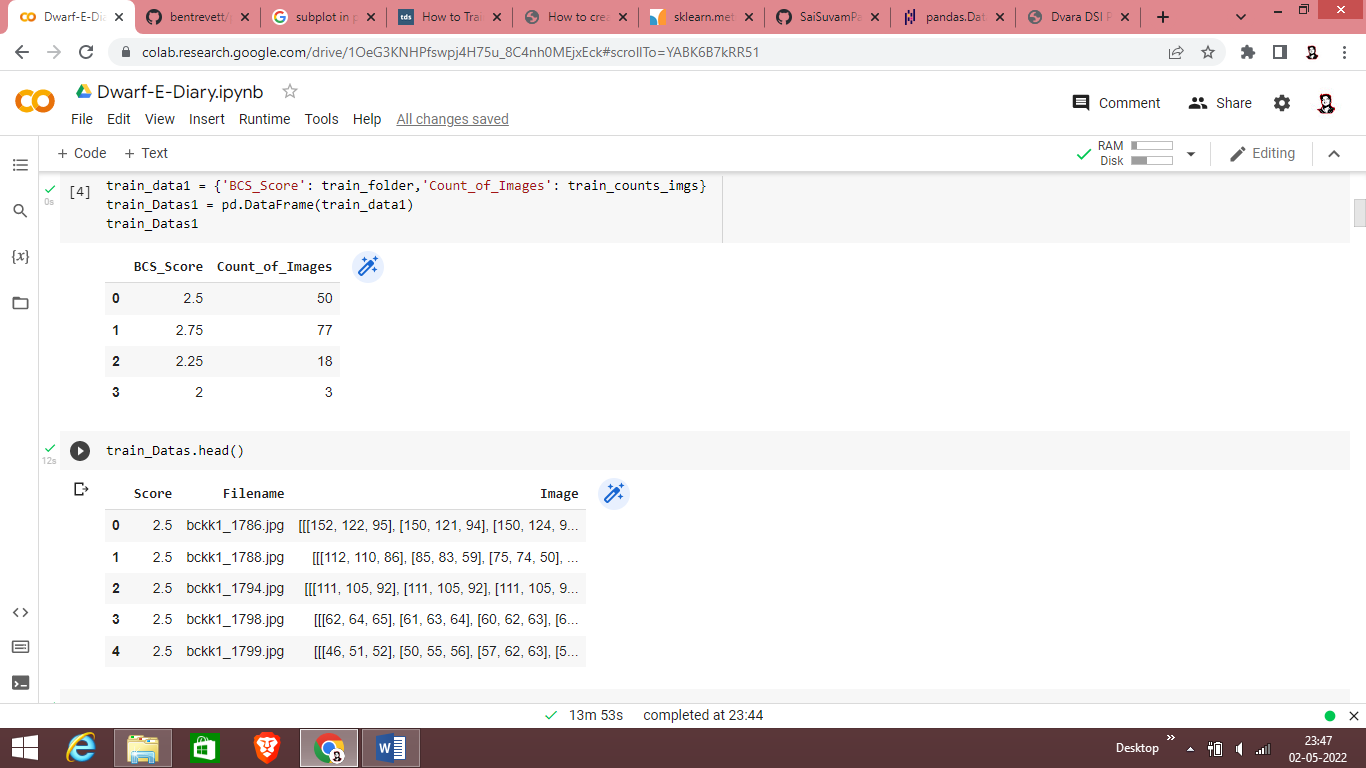
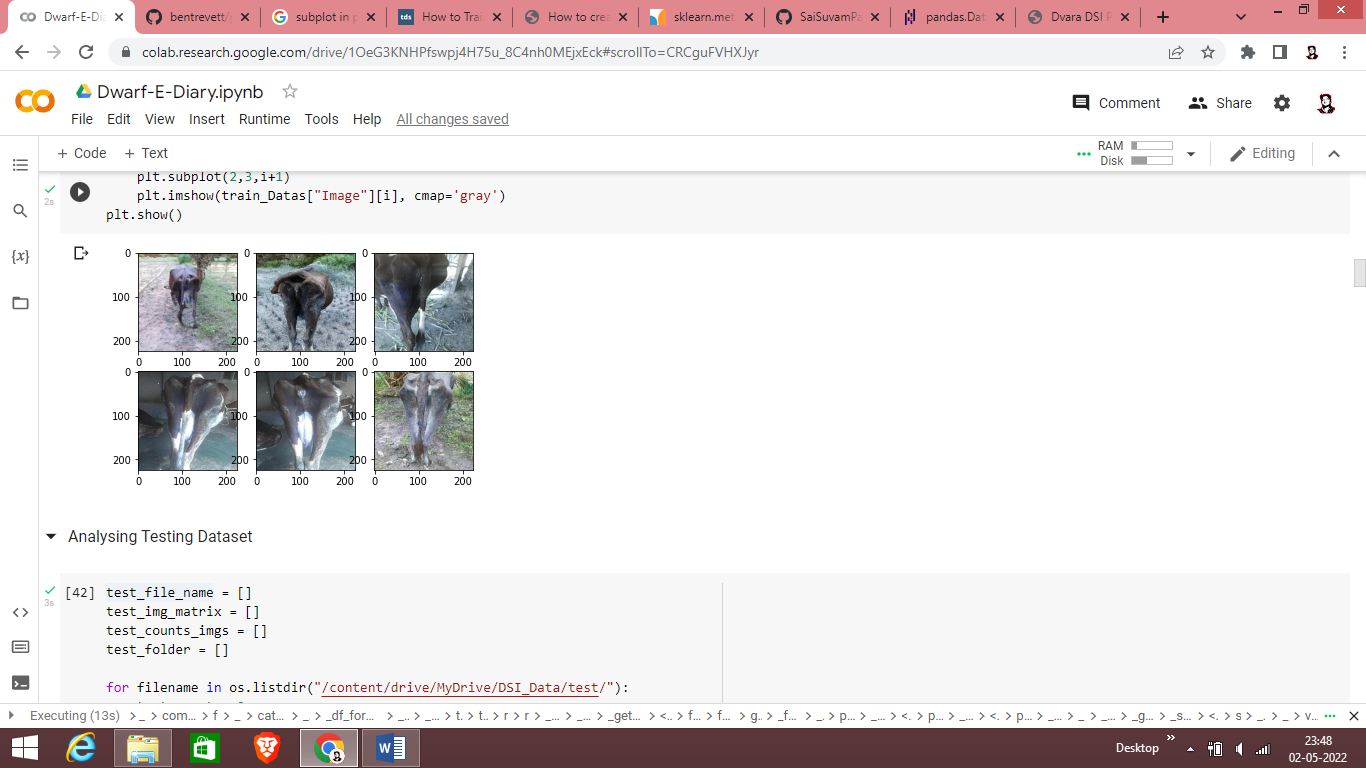
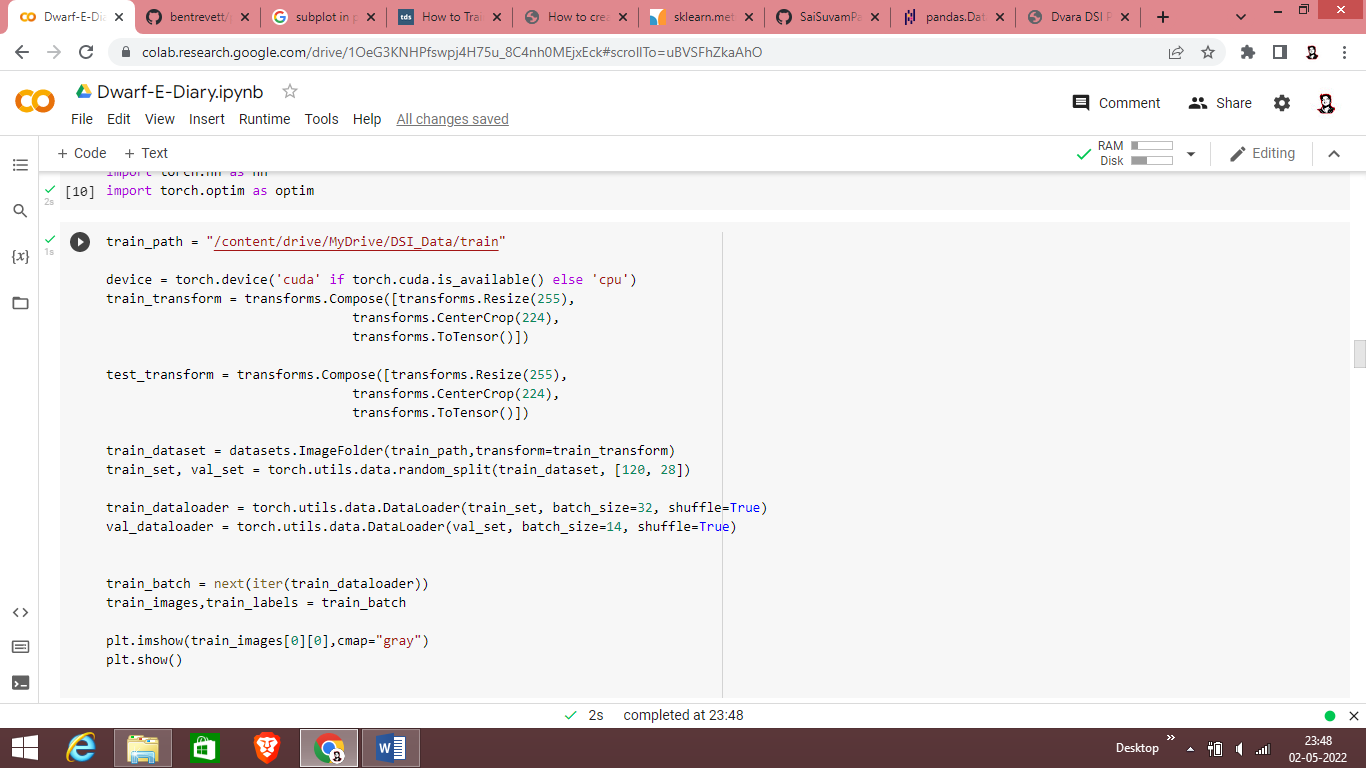
**Body Condition Score**

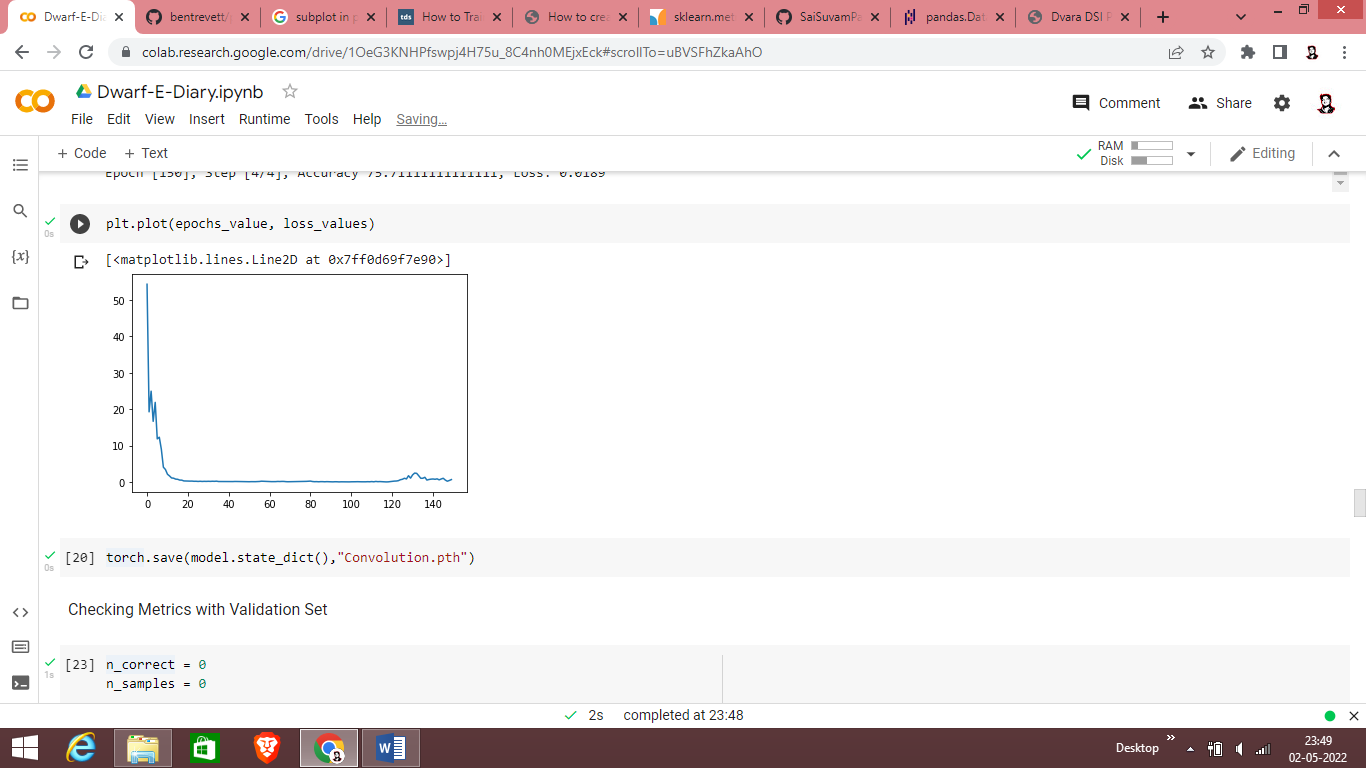
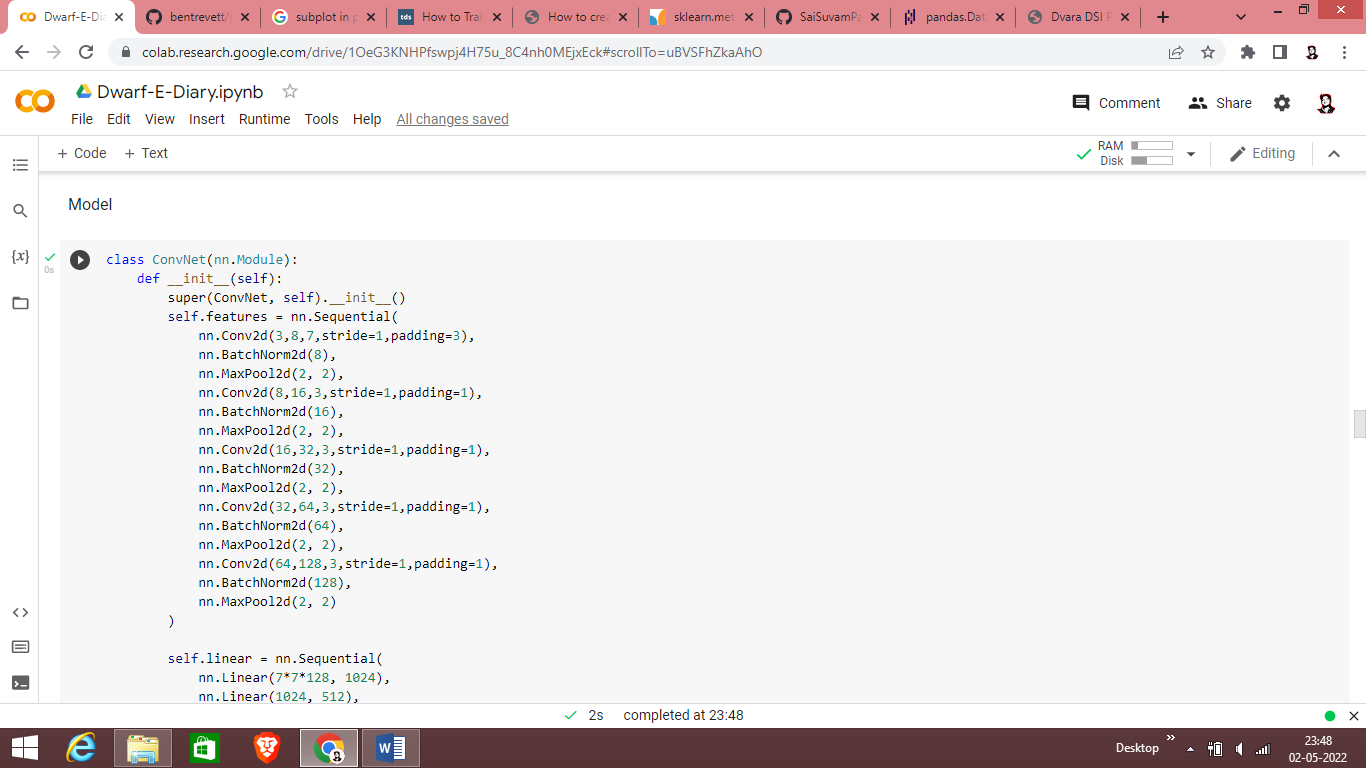
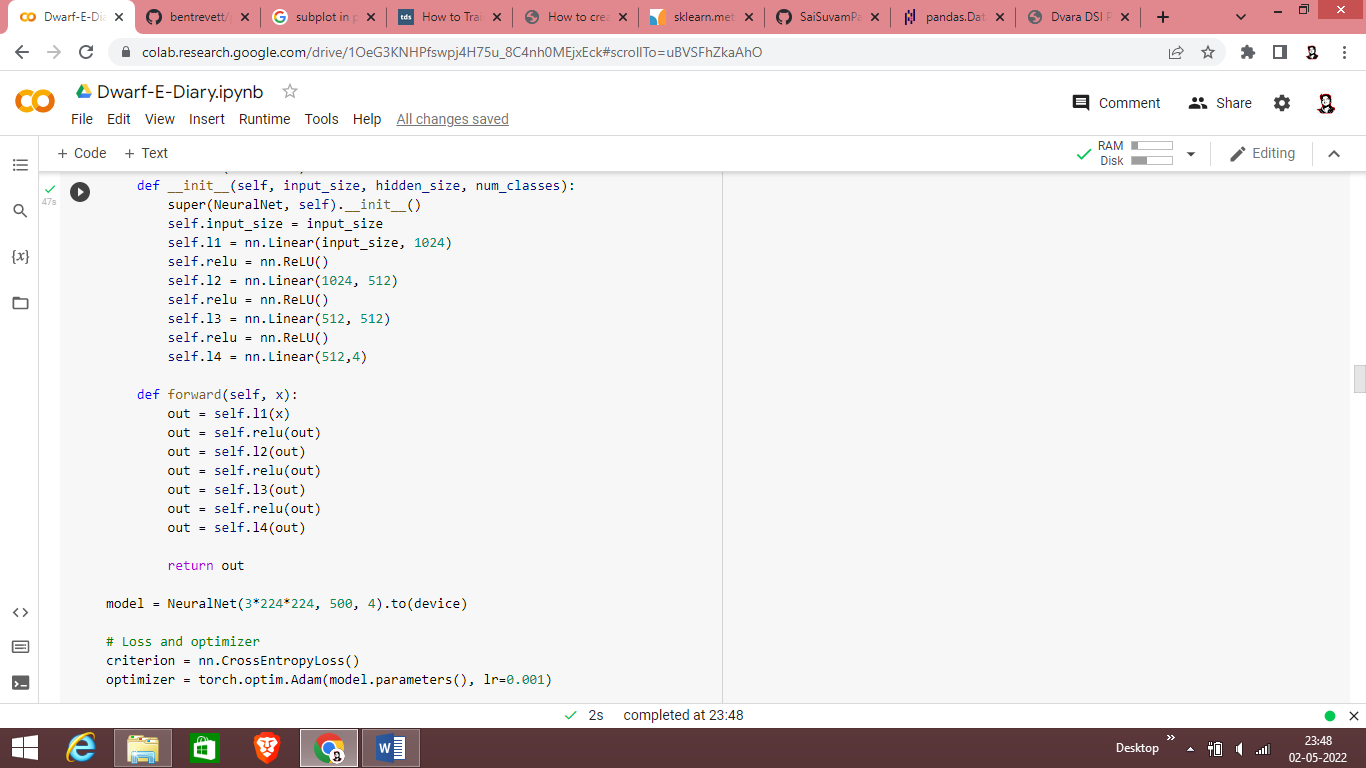
So At first talking about the Dataset Provided, The DSI Dataset consisted of 2 folders “Train” and “Test”. The train folders contains 4 subfolders each assigned with a BCS Score of 2, 2.25, 2.5, and 2.75.



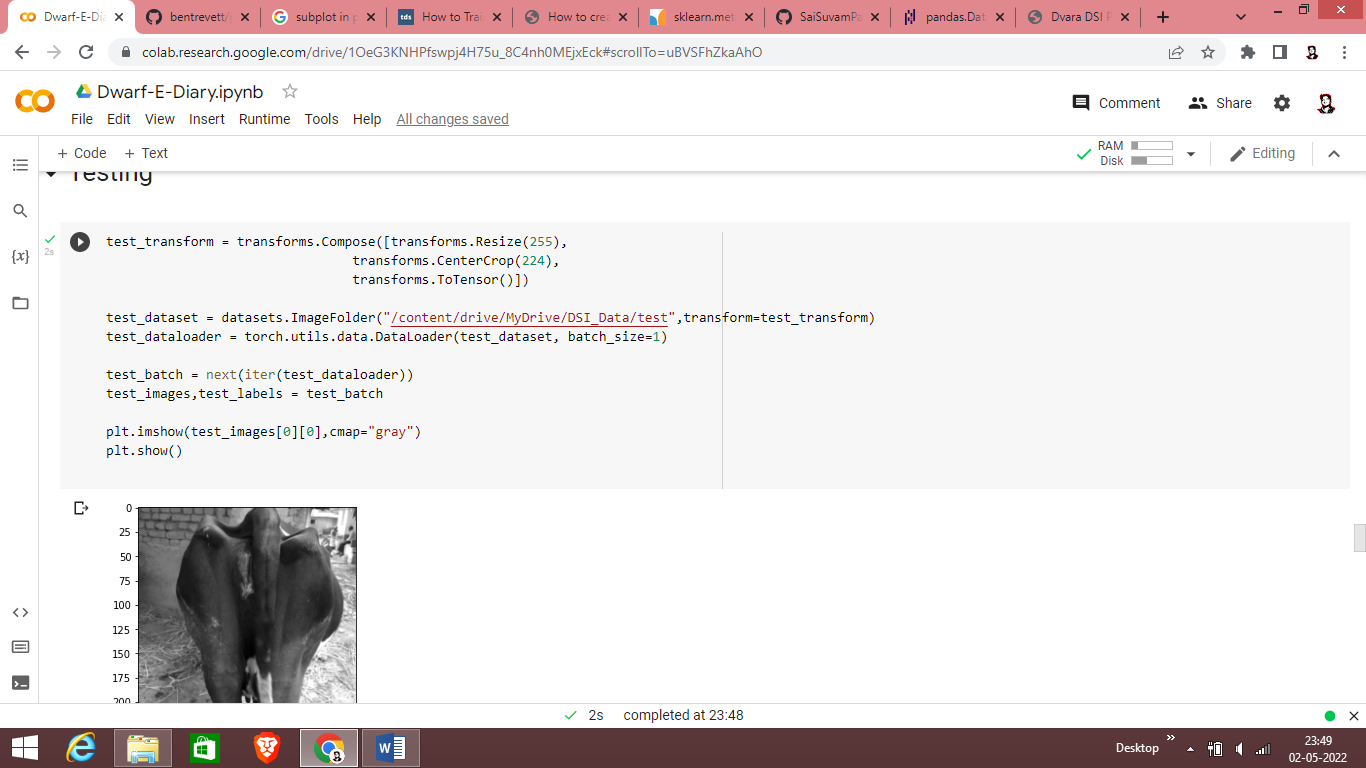
The total no of images in the training folder and testing folder is 148 and 26 respectively.







I imported the datas using the imagefolder function of the dataset module and performed various transformation over it. The images present in the data folders were not uniform sized so I resized it to 255 X 255 and then Center cropped it to focus on the cattle part and then converted it to tensor. Then the datas were loaded for further computation using the dataloader which basically returns the iterative dictionary of the image matrix and its labels. Now talking about the model, I implemented multiple models for training the dataloaders. Initially starting with the pretrained Resnet18 model, I started tuning the hyperparameters of it liking changing the learning the learning rate to 0.01 to 0.1 to 0.001, changing the momentum value to a range of 0.6 from 0.9 and changing the optimisers like the sgd, adam, gd but due to a small number of training samples the accuracy fluctuated around 12-15 and the loss was even too high which was not suited for training our model. The resnet model used above was the pretrained one where the model’s parameter was not at all trained in the back propagation and the default weight values were only used. So I decided to once try training the weights of the resnet18 model and tune the hyperparameters and then start the training but still it didn’t show much improvement and the accuracy varied between 15-25. So I shifted my decision to Neural networks and try the fully connected layer to train my datas and defined 4 linear layers with a relu activation function within the layers to make non linear transformations. The loss function used to find out the residual is CrossEntropy and I used Adam optimiser to update the weights of the Neural network, Learning rate value is set to 0.001. Each image’s matrix and labels were accessed from the dataloader and passed into the model, The output we get is compared with the ground value and loss is calculated and the entire model is backpropagated which trains the model. Here In this procedure we got an accuracy of around 40% which is still not acceptable and the Linear layer network didn’t suit well for our dataset. So I implemented another network layer which consisted of 5 sets of Convolution layer, Batch Normaliser and Maxpooling layer and the result of this is then passed into 3 fully connected layers for predicting the probabilities of the input provided to it. Cross Entropy was used as loss function and the adam optimiser was used for back propagation. This layer of networks worked well for our dataset as with a epochs of 150 we got an accuracy of 60%-65%. The dataset was splitted originally into training and validation and the above mentioned accuracy was for the training set and testing for the validation set I got an accuracy around 40%-45% for each batch I divided the validation set for.



Finally after Defining the model I tested for the testing dataset and performed the same computation and transformation over the testing images and loaded the datas using dataloader and passed it into our model. The output that I got was then appended into a dataframe with the testing images filename mapped with it. The panda dataframe was then exported as a excel file and was saved in our local device.

