**Assignment-3**

**Srushti Wadekar – 1102484**

**Notes for Execution**

1. Keep the dataset folder i.e. “DS” in the same folder as the python and model files in zip folder.
2. For src, set the path of the above folder, i.e. which contains the dataset.
3. For running the saved model, skip execution of cell numbers from 5-10, 17, 18, 22-26.

**Dataset**

The dataset that is been used for this assignment is image dataset. It has images of

1500 CT scans of various parts of the body.

**Preprocessing Steps**

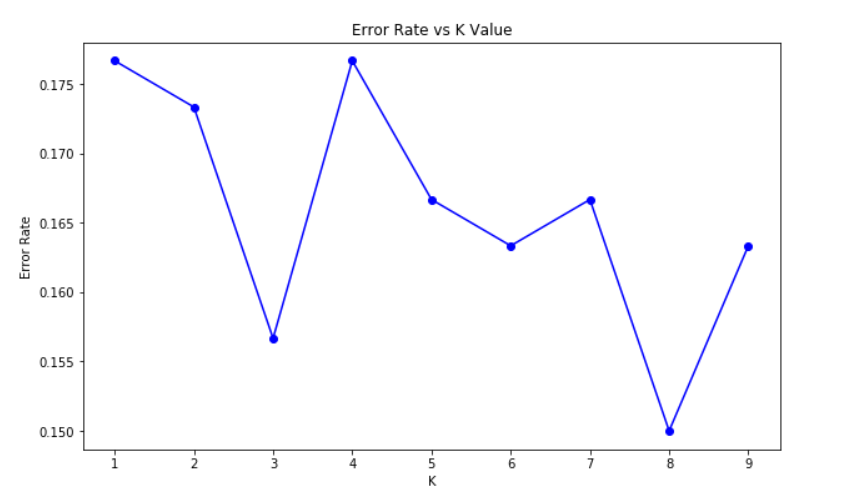
1. The images have been read using OpenCV library.
2. After reading, images have been resized into size of (32,32) using OpenCV and stored in a numpy array.
3. Data augmentation is been implemented on the training dataset before passing it to CNN network.
4. Data augmentation is done for generating more samples of data, as the original dataset is less in size for training a model.
5. For data augmentation, a folder of training images is created.
6. Data augmentation is done using image generator function on the training folder.
7. The augmented data is passed to CNN model by considering type of images as rgb.

**Feature Extraction**

1. The feature extraction of images is done using CNN network.
2. A model of 2 convolution 2D layers by applying batch normalization, and two dense layers have been used to extract the features from the network.
3. 10 features are extracted from the second last dense layer of the CNN network.
4. The features from the network are extracted by training the network on the augmented training data.
5. Training and testing features are extracted separately and stored in respective numpy array.

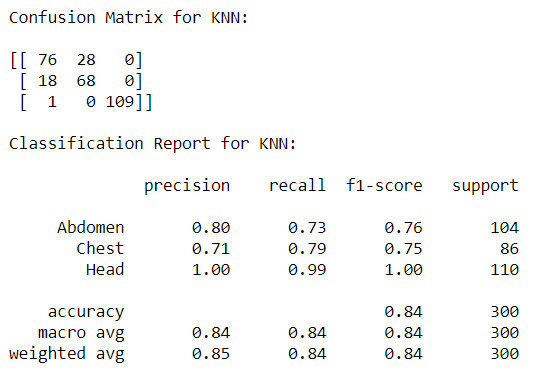
**KNN model Results**

1. The library used for implementation of KNN is sklearn library.
2. KNN was tested over different values of neighbours between 1-10 to evaluate the minimum error rate.
3. The below figure shows the graph of error rate generated for different values of k or neighbours.



*Fig.1 Error rate generated for k values from 1-10*

1. From the above graph, k=3 and k=8 showed a very low error value. Hence, k=3 was selected for training the KNN model over extracted training features.
2. Training KNN on k=3 gave accuracy of 85% on the testing set.
3. The below given figure shows the confusion matrix and classification report of KNN model.

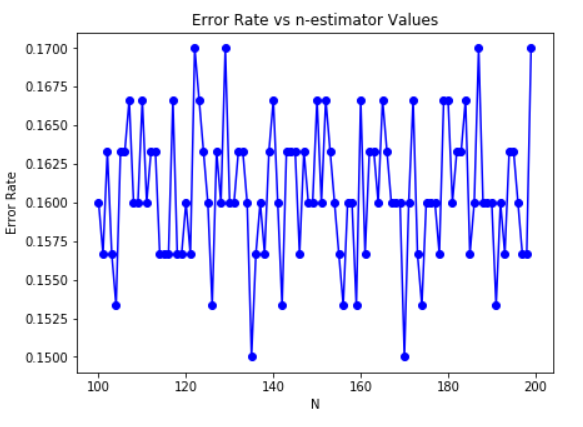


*Fig.2 Confusion matrix and classification report of KNN*

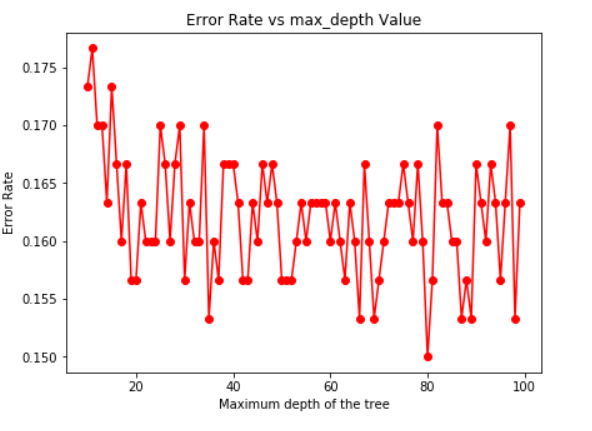
1. The training accuracy of the model was greater than it’s testing accuracy. Hence, the KNN model performed descent on classifying the images.

**Random Forest model Results**

1. The library used for implementation of Random forest is sklearn library.
2. Hyperparameter tuning was done before training the optimal model by using random search method.
3. Hyperparameters that were considered for tuning are the n-estimators, maximum depth and bootstrap.
4. For random search, n-estimators withing range of 100-200 and maximum depth within the range of 10-100 was considered.
5. The below graphs show the error rate by considering different values of hyperparameter for testing.

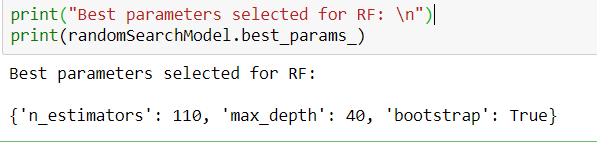


*Fig.3 Error rate for different values of estimators*



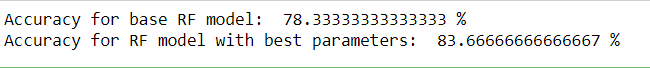
*Fig.4 Error rate for different values of maximum features*

1. The random search for hyperparameter tuning was done using 10 iterations and 3-fold cross-validation.
2. The best parameters that were selected for random forest are shown in the below figure.



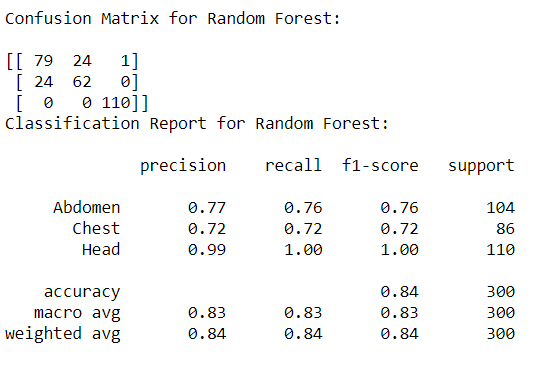
*Fig.5 Best parameters for Random Forest using random search*

1. After training the base model and model with best parameters, accuracy on the testing set was increase which is shown in the below figure.



*Fig.6 Comparison of accuracies for random forest*

1. The below given figure shows the confusion matrix and classification report of random forest model based on optimal hyperparameters.



*Fig.7 Confusion matrix and classification report of Random forest*

1. The training accuracy of the model was greater than it’s testing accuracy. Hence, the random forest model performed descent on classifying the images.

**Conclusion**

By comparing the results of both the model, KNN performed slightly better compared to random forest. The difference between both the model was very less which was around 1-2%. Random forest would have outshined more if more than three parameters were considered during hyperparameter tuning. Even though the training accuracy was greater than testing accuracy in both the models, there was a huge gap between the training and testing accuracies. Still cannot prove that the model was overfitting just by considering the difference of accuracies. Hence, according to the given experiment both the model performed descent on the given dataset.