Assignment - 3

Nishchay Trivedi, Student id: 1106924

Pre-processing Steps:

Initially all the files are stored in one array from the datasets given. Later, to get the target value I have checked the file's first character and if the first character is A then value is given as 0, if it is C then value is 1 and if it is H then value is 2:

To read the images opencv is used and to pre-process the images I converted them into grayscale and scaled the pixel value. Then the image is resized in 32 x 32 using opencv's resize function. Then for further preprocessing I performed the data augmentation using keras data generator function. After that each image is converted to a numpy array. So the dimensions of the image will be 1500x32x32 and for the target variable it will be 1500x1. Then the images are split into train and test. After that, to get the features CNN model is built which is shown as below:

```
Model: "functional_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3_input (InputLayer)  [(None, 32, 32, 1)]       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 64)        1664
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 12, 12, 32)        18464
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 6, 6, 32)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 4, 4, 16)          4624
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 2, 2, 16)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 64)                0
_________________________________________________________________
feature_dense (Dense)        (None, 100)               6500
=================================================================
Total params: 31,252
Trainable params: 31,252
Non-trainable params: 0
_________________________________________________________________
```

Fig 1

To train the mode the dimension is expanded and shaped to 1500x32x32x1. Then extracted the image(X) and target(Y) features. Then to evaluate them kNN and random forest is used with hyper-parameter tuning.

Graphs:

K values comparison: in figure 2, the performance is plotted with different values of k [3,5,7,9]. For the performance matrix I have used accuracy, precision and F1 score.

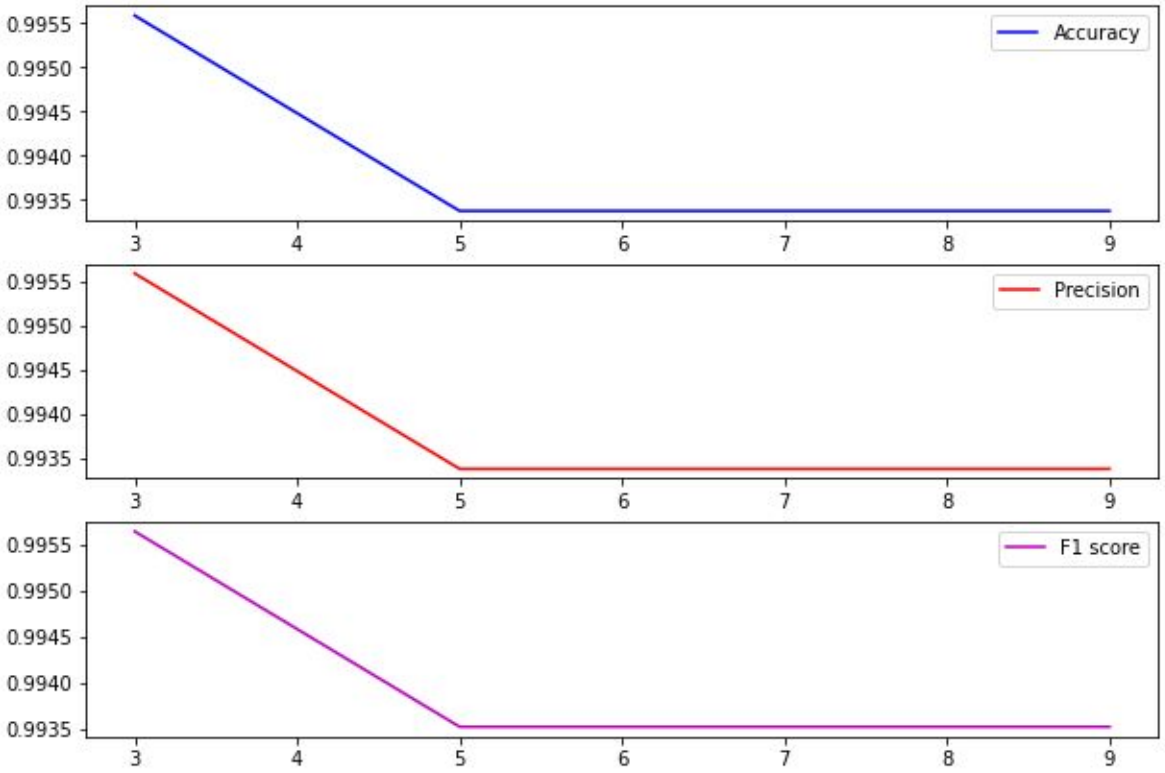

Fig 2

Confusion matrix for kNN: After evaluating the kNN, the result of confusion matrix as shown in figure with target names:

Figure 3

Here in kNN I have tuned different values of k ranging from 3,5,7,9. And thes best I got is 3.

- Confusion matrix for RF: After evaluating the RF, the result of confusion matrix as shown in figure with target names:

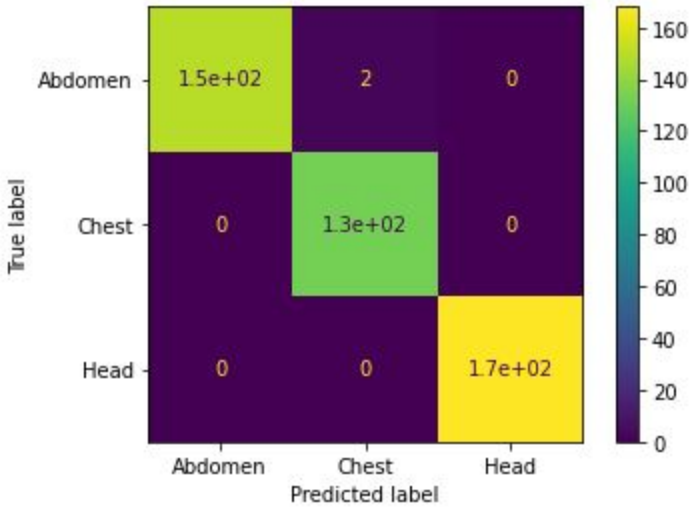

Figure 4

Here for the hyper parameter tuning, I have tuned 3 parameters, n_estimators, bootstrap and depth and the best value I got is 200, False and 20 accordingly.

Performance:

Result of kNN:

```
Accuracy  : 0.9955849889624724
Precision : 0.9956503965333988
f1Score : 0.9955868436424081
```

Result of RF:

```
Accuracy  : 1.0
Precision : 1.0
f1Score : 1.0
```

As from the result it is clearly seen that it almost gives the best performance (may overfits) . The result of this can be data is not processed in a good manner or it is straightforward to predict the result. To test the actual performance, the images can be processed more using proper data augmentation technique or some other techniques like threshold, transformation, smoothing etc.