CMPE255-Project Report PR2

Rohan Kamat
013759252

Multiple approaches were used to complete the assignment.Each algorithm and approach produced a variety of result.The most favorable result was generated by ***KNN (neighbours=5) along HOG,with the F1 score of 0.902***.

Approach 1: Glob a python library was used to iterate through the directory containing the images, the images were visited in a random order not in chronological order, because of which the sorting the images became necessary, to match the labels.The calcHist library from cv2 was used to calculate the histogram of each image and store it in the list called training.The names and the data from list were used to form a dataframe which was then sorted by names.The similar approach was for testing data.The class labels were read from the file and stored in the list called labels.*RandomForestRegressor* was used to model with *n_estimators=100* and *random_factor=42*.The result was a 0.46 F1 score.The reason being the data wasn't balanced and biased towards particular class.

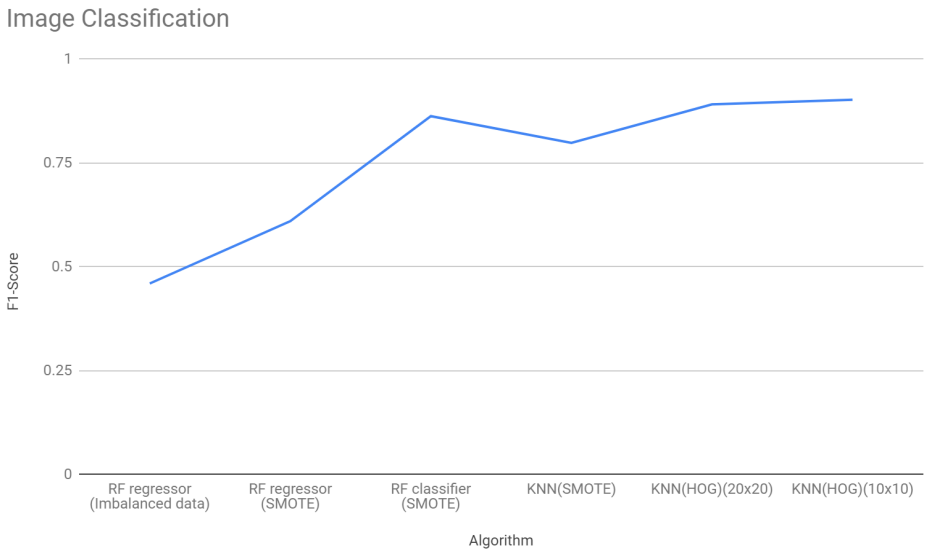
Approach 2: On searching methods to balance data, I came across *SMOTE (Synthetic Minority Over-sampling).SMOTE* generates new samples that are coherent with the minor class distribution.This made all training set unbiased. *KNN was used with neighbours =100 which gave 0.79,Gradient boosting -0.52,RandomForestRegressor- 0.72.*On realizing not much improvement in F1 score, another method was used called as *weighted RandomForestClassifier.*

$$w_j = \frac{n}{kn_j}$$

The classes are automatically weighted inversely proportional to how frequently they appear in the data.This method gave a F1 score of 0.86 which was subsequently higher.

Approach 3: HOG(Histogram of Oriented Gradients) was used, to standardize the images were resized to 64 by 64 and converted to grayscale.The images were passed through these functions and similar procedure as stated in the first approach was used.*The pixels_per_cell were kept as 20,20 which produced an F1 score of 0.89.On similar approach with pixel per cells 10,10 produced an F1 score of 0.902 which was the best score of all approaches.*

To illustrate the use of dimensionality reduction PCA was used from scikit-learn library, deployed as a function.On integrating it with the current method, the F1 score reduced by 0.1, so wasn't used just stated.A total of 16 submissions were made and the relevant ones are plotted as follows.



F1 score	**Rank**
0.9021	7