# IMAGE CLASSIFICATION: A COMPARATIVE STUDY

***Name: Shrey Bhatt***

## Image Classification Problem

The definition of image classification problem: Given a set of images with their labels, construct a model which can accurately predict the label of a given image i.e. classify the image based on the features inside the image. It is a problem that comes under the category of supervised learning and so labels must be initially provided to the images to train the model to identify features with respect to labels.

A dataset is provided named 'tentest classes':

- The dataset contains 99 images.
- The images are the top views of different kinds of landscapes.
- The images are divided into 10 classes like agricultural, airplane, beach, freeway, golfcourse etc.

## Feature Extraction

Features, in the context of this problem, can be defined as the data retrieved from each of the image in order to train your model. They are the driving factor to determine the accuracy of the model. They are used to represent the image in various possible ways. They may be real valued or categorical in nature.

Using only a single type of feature can reduce the prospects of better results since a single feature like color may not be able to well distinguish between different classes like in case of fruits where different type of fruits may have the same color. On the other hand if we use too many feature sets, computation may become inefficient. Here we have used three types of features:

- Pixel values: It represents the RGB values of pixels of the image, which represents the expected pixel value with respect to position of an image of a certain label.
- Color Histogram: A color histogram represents the distribution of various colours for an image. Color is very important feature for an RGB image. Here, histogram is represented using HSV representation and frequencies are calculated for every possible values.

- Hu Moments: A moment is the weighted average of intensities of pixels. Hu Moment is the collection of seven function of such intensities. They are used to determine area of objects inside the image.

To extract the features of the images, the directory is set and all the image files are listed. Then each file is iterated and then using pillow library the pixel values of image stored into numpy array.

1. Here we will divide the image into blocks such that blocks will form a dimension of 3*3 to represent the image. To do this, initially the height and width of image are extracted and then they are divided by 3 to get the height and width of individual blocks. Then nine individual blocks are extracted based on the position and considering various end cases like the boundary pixels so that every pixel is properly assimilated. Then from that sub matrix, summation of R, G and B values are done and using that mean is calculated. So by this method we create a feature set of 27 values containing average red, green and blue pixel values of image divided into 9 blocks. This gives further advantage since it reduces computation but may also lead to data loss.
2. Using openCV, the RGB pixels are converted to HSV format and returned to a new list. Then using calcHist method of cv2, we create a histogram of the values of HSV list. Setting the bins parameter to 8, the frequencies, entire range is divided into 8 segments and frequencies are calculated accordingly. Finally, using the flatten() method, the list is converted to a one dimensional list. So, by this method a list of 512 features is obtained describing the HSV attributes of the image.
3. Again using openCV, the RGB pixels are converted to grayscale representation and passed to HuMoments method in order to get the HuMoments value to represent intensity as described above. It returns a list of 7 values.

Finally, we combine all these features in order to form a final featureset consisting of 546 attributes. To get the class value, the filename is parsed and used to get proper word used to act as label. The process is repeated for all images.

Thus, this is how feature set and class values are constructed for the entire image dataset.

## Models Used for Classification

Using sklearn, classification models are used in order to solve the problem. The following models have been used to train and test the data for classification:

1. Logistic Regression
2. Gaussian Naïve Bayes
3. Decision Tree Classifier:
    a. Using entropy criterion
    b. Using gini criterion: Here, variation in maximum depth of tree is carried out to observe the optimum depth in order to maximize accuracy by avoiding over fitting.
4. Using SVM classifier
5. Using Bagging ensemble over Decision tree constructed by Gaussian classifier.
6. Performing K-Folds cross validation technique over above models.

## Model performance and evaluation

After deciding the models, test data and train data was prepared from the data set. Using that, all models were trained and tested. The models were obtained using library sklearn. The object of model was obtained by using constructor and passing appropriate parameter values. Using the fit() function of the model, model is trained by passing the training data and labels. Finally using test data and test labels, the accuracy was calculated using score() method and it's confusion matrix is printed using confusion_matrix() method. Using the predict method, a sample file is tested and output is compared against the actual value.

1. Logistic Regression:

The output obtained using Logistic Regression is as follows:

```
Using Logistic Regression, Accuracy: 0.933333333333
File: tentest classes\beach05.tif   Actual:  beach   Predicted:  beach
Confusion matrix
[[3 0 0 0 0 1 0 0 0 0]
 [0 3 0 0 0 0 0 0 0 0]
 [0 0 4 0 0 0 0 0 0 0]
 [0 0 0 5 0 0 0 0 0 0]
 [0 0 0 0 4 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0]
 [0 1 0 0 0 0 0 3 0 0]
 [0 0 0 0 0 0 0 0 3 0]
 [0 0 0 0 0 0 0 0 0 1]]
```

It showed high accuracy and also predicted correctly against a sample test data.

2. Gaussian Naïve Bayes

```
Using Gaussian Naive Bayes, Accuracy: 0.866666666667
File: tentest classes\beach05.tif   Actual:  beach   Predicted:  beach
Confusion matrix
[[1 0 0 0 0 0 0 0 0 3]
 [0 3 0 0 0 0 0 0 0 0]
 [0 0 4 0 0 0 0 0 0 0]
 [0 0 0 5 0 0 0 0 0 0]
 [0 0 0 0 3 0 0 0 1 0]
 [0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 4 0 0]
 [0 0 0 0 0 0 0 0 3 0]
 [0 0 0 0 0 0 0 0 0 1]]
```

Though it gives comparatively low result than Logistic Regression, predictions are mostly accurate and so it is also a better prospect for this problem.

3. A Decision tree using Information Gain

```
Using Information Gain Decision tree, Accuracy: 0.7
File: tentest classes\chaparral04.tif   Actual:  chaparral   Predicted:  chaparral
Confusion matrix
[[3 1 0 0 0 0 0 0 0 2]
 [0 1 0 0 0 0 0 0 0 0]
 [0 0 4 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 2 0 0 0 2 0]
 [0 0 0 0 0 3 2 0 0 0]
 [0 0 0 0 0 0 2 1 0 0]
 [0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 3 0]
 [0 0 1 0 0 0 0 0 0 1]]
```

So, accuracy is quite satisfactory and model may improve on providing more instances for training.

3. B Decision tree using Gini Index

```
Using CART Decision tree, Accuracy: 0.766666666667
File: tentest classes\overpass07.tif   Actual:  overpass   Predicted:  overpass
Confusion matrix
[[5 0 0 0 0 0 0 0 0 1]
 [0 1 0 0 0 0 0 0 0 0]
 [0 0 3 0 0 0 0 0 0 1]
 [0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 2 0 0 0 2 0]
 [0 0 0 0 0 3 1 1 0 0]
 [0 0 0 0 0 1 2 0 0 0]
 [0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 3 0]
 [0 0 0 0 0 0 0 0 0 2]]
```

Here, also accuracy is high enough. Here variations were also performed by setting different values to depth, but it was observed that it did not contribute much to the accuracy, so it is advisory to let the model construct the entire tree for the given dataset.

4. SVM classifier:

```
Using SVM classifier, Accuracy: 0.0666666666667
File: tentest classes\airplane09.tif   Actual:  airplane   Predicted:  storagetanks
Confusion matrix
[[0 0 0 0 0 0 0 0 0 4]
 [0 0 0 0 0 0 0 0 0 4]
 [0 0 0 0 0 0 0 0 0 2]
 [0 0 0 0 0 0 0 0 0 2]
 [0 0 0 0 0 0 0 0 0 2]
 [0 0 0 0 0 0 0 0 0 3]
 [0 0 0 0 0 0 0 0 0 5]
 [0 0 0 0 0 0 0 0 0 4]
 [0 0 0 0 0 0 0 0 0 2]
 [0 0 0 0 0 0 0 0 0 2]]
```

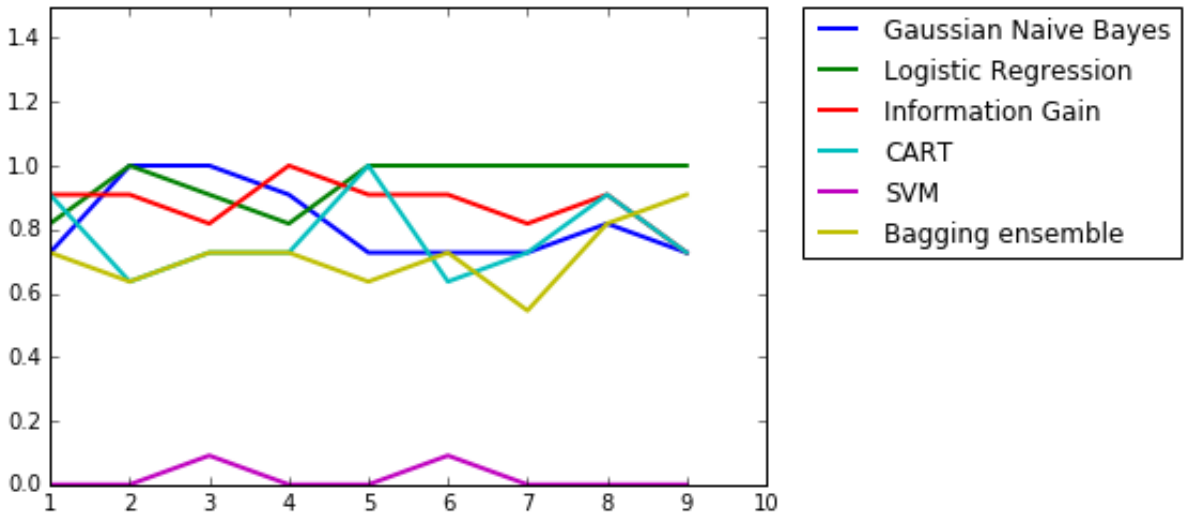
As shown, SVM provides very low accuracy and also cannot make correct predictions. So it is not preferable for given problem.

5. Bagging ensemble:

```
Using Bagging Ensemble in CART, Accuracy: 0.733333333333
File: tentest classes\storagetanks00.tif   Actual:  storagetanks   Predicted:  storagetanks
Confusion matrix
[[4 0 0 0 0 0 0 0 0 0]
 [0 2 0 0 0 1 0 0 0 1]
 [0 0 2 0 0 0 0 0 0 0]
 [0 0 0 2 0 0 0 0 0 0]
 [0 0 0 0 2 0 0 0 0 0]
 [0 0 0 0 0 3 0 0 0 0]
 [0 0 0 2 0 0 3 0 0 0]
 [0 0 0 0 0 2 0 2 0 0]
 [0 0 0 0 1 0 0 0 1 0]
 [0 0 0 0 1 0 0 0 0 1]]
```

Over certain results, it was found that Bagging ensemble over CART decision tree improves its accuracy over a slight extent. It is also a preferable model for this problem.

6. Finally, after all the individual modelling, all models were given to input under K-Folds cross validation technique to observe their consistency over iterations. Using sklearn, an instance of kfold is obtained by passing the number of samples, the number of folds and the random state. Then, the object is iterated over to get train indexes and test indexes using which above procedure is followed for different models. Accuracy of models over all iterations are obtained and displayed. The result obtained is as shown:




## Comparative study and analysis of above results:

Hence, the accuracies given by the models are:

| Model | Accuracy |
|---|---|
| Logistic Regression | 93.333% |
| Gaussian Naïve Bayes | 86.66% |
| Decision Tree using Information Gain | 70% |
| Decision Tree using Gini Index | 76.66% |
| SVM Classifier | 6.66% |
| Bagging ensemble using Decision tree: Gini index | 73.33% |

So it is clear from above results that Logistic Regression is giving constant well performance over the iterations. Also other models like Gaussian Naïve Bayes, Information Gain are providing almost equal output for accuracy. The reason for well performance of Logistic Regression may be due to consideration of each class separately since it can deal with only binary labels at a time and produces one or zero accordingly. So it considers whether the given image is present in the considered label. If yes, it gives one otherwise it repeats the same process by discarding only that label and continuing with others.

Also, Gaussian Naïve Bayes is designed to work well with numerical data by performing statistical analysis, which may be helpful in images which look almost similar and so it also provides fruitful results.

CART decision tree and Bagging ensemble may produce fruitful results on further study of parameters or providing more data for training.

SVM classifier gives low accuracy value over all iterations and so it is less preferable. It may be due to increasing complexity due to involvement of too many dimensions.

Further, some features may be added or modified in order to increase the efficiency.

## Conclusion:

Thus, this was a comparative study of the image classification problem with features and models as described. Better and different results may be produced on selection of different features and models, and so this problem has attracted attention of many researchers and enthusiasts.