CS 256: Homework3

SJSU ID: 012447851

Name: Mayuri Wadkar

Instructions to run the program:

Command for execution from command prompt:

```
$ python Main.py
```

My python implementation requires following external libraries:

1. numpy==1.13.1
2. Pillow==2.5.1
3. matplotlib==2.0.2

(Note: Please use above versions, as older version would result in errors.)

Folder structure should be as mentioned below:

1. Place images to classify in "./ImagesToClassify" folder
2. Place source flag and target flag images for evolving a new flag in "./EvolveOneFlagToAnother" folder
3. Place flags to cluster in "./flags" folder
4. Place training data for Knn and 3-fold cross validation in "./trainingData" folder inside two separate directories "./trainingData/headshots" and "./trainingData/headshots"

First program run creates and stores data in external files as shown below. These files won't be present on your first run and will be created in the same folder where Main.py resides. For further runs, data will be fetched from these files. Deleting these files would result in slow performance, as the data will be computed again.

System Design:

System provides following options to user:

1. Classify image of your choice:

In order to decide a class label for an image, environment extracts following features from every image:

a) Skin colored pixel percentage

Headshot images have significant number of pixels with skin color. I have used formula mentioned below to decide if a pixel is skin colored or not. [1]

```
if R > 95 and G > 40 and B > 20 and (x - y) > 15 and ((10 * z) > 15) and R > G and R > B:
   skinColorPix += 1
```

b) *Blue colored pixel percentage for identifying water and sky in landscape images*

Majority of landscape images can be distinguished by seeing presence of sky or water. Categorizing pixel as a shade of blue and taking percentage of total number of blue shade pixels in an image would solve the problem. Following formula has been used to decide if a pixel is blue shaded or not. [2]

```
if B > (1.2 * max(G, R)):
   skyWaterPix += 1
```

c) *Green colored pixel percentage for identifying greenery in landscape images*

Green shade, representing greenery, is seen as the most dominating color in landscape images. Percentage of green colored pixels, have been used as one of the feature in this system.

d) *Sunset color pixel percentage for identifying sunset presence in landscape images*

After analyzing landscape images, significant number of landscape images have a sunset or sunrise moment captured in it. Their color shades would help in classifying an image as a landscape.

e) *Blurriness percentage*

After analyzing headshot images, I noticed that for most of the headshot images, backgrounds have been blurred. This feature if captured well would help improve accuracy of a classifier. Percentage of blurriness in image would play a significant role.

A pixel can be categorized as a blurred or non-blurred pixel by comparing its energy with the mean energy of an image. [3] Face part of an image is a well-focused part, where pixels have high energy. This high and low energy areas can be used to find the blurriness percentage.

All the above features contribute to a feature vector associated with an image. Using this percept vector, adjustable k value and Euclidean distance metric, classifier assigns appropriate label to images.

2. 3-Fold Cross-validation on landscape and head-shot data-sets

System automatically creates 3 validation and 3 training datasets, one pair for each round. It randomly partitions the data in folds and computes accuracy for multiple k values (1,3,5,7,9). It also plots the graph for K vs accuracy percentage.

3. K-means clustering on landscape and head-shot data-sets

K means clustering has been implemented for k = 2 i.e. number of clusters = 2.

4. Hierarchical clustering on landscape and head-shot data-sets

Single link hierarchical clustering has been implemented.

5. Cluster flag images of all sovereign countries

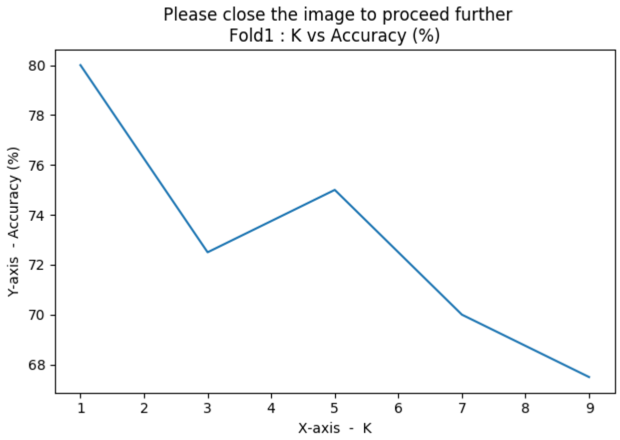
Single link hierarchical clustering has been implemented.

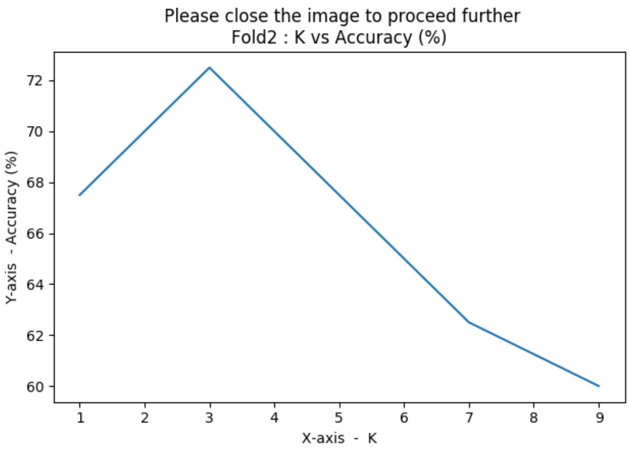
6. Evolve one flag into another

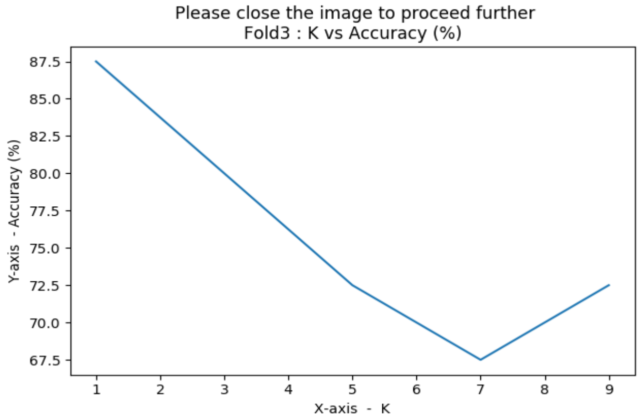
Genetic Algorithm [4] has been used to mutate bit strings in an image. Quality of result depends on number of iterations.

Experiments and results:

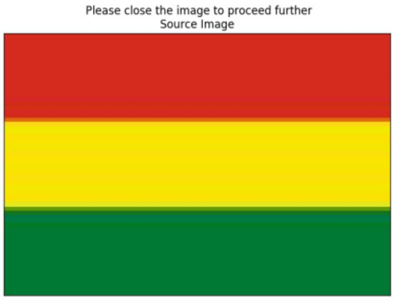
Result of 3-fold cross validation:

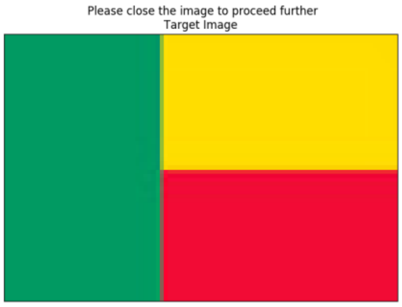


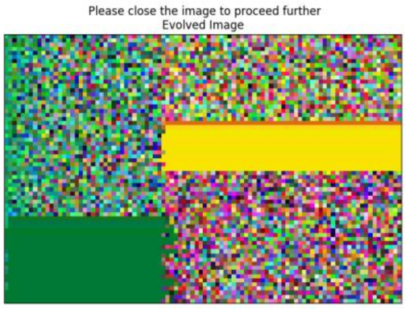




Result of evolving one flag into another:







Performance Improvement and future work:

Every time a program is run, data gets loaded for Knn classification, K-means and 3-fold cross validation and it gets written to an external file. For all further runs, the program will use the data from external file unless the file has been physically removed from the project root directory. Validation and training data is fetched from external file for each round except the first run when the data gets written to external file.

To increase accuracy of an Image classifier, features need to be extracted carefully taking into consideration the subject. One must look for obvious features, which humans would pick up to distinguish images. For example, for headshot image, we can extract binary value feature – triangle (eye, eye, mouth) detected. For landscape image, if a sun exists in an image, we can add binary value feature - hot spot detected.

References:

1. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.521&rep=rep1&type=pdf
2. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.1294&rep=rep1&type=pdf
3. https://www.researchgate.net/post/Is_there_any_particular_method_to_check_the_blurriness_of_an_image
4. http://www.ai-junkie.com/ga/intro/gat2.html