UNC CHARLOTTE

PROJECT REPORT

ON

AUTOMATIC IMAGE CLUSTERING

Submitted By
Gazal Chawla (800930562)
Manasa Anil Kumar (800900728)
Rachapudi Venkata Vaishnavi (800904888)

VD Project Write-Up

Program Statement

Automatic Image Clustering:
Download 1 million images and extract SIFT features of each. Apply AP clustering technique to get an optimum partition of the images according to their visual similarity.

Background

The four concepts used in the program are SIFT, Bag of Words, Hierarchical clustering and AP Clustering.

- **SIFT (Scale Invariant Feature Transform)** is an algorithm that detects and describes local features in images. SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

- **Bag of Words** is a vector of occurrence counts of a vocabulary of local image features. To represent an image using BoW model, an image can be treated as a document. Similarly, "words" in images need to be defined too. To achieve this, it usually includes following three steps: feature detection, feature description, and codebook generation.
  Feature representation
  After feature detection, each image is abstracted by several local patches. Feature representation methods deal with how to represent the patches as numerical vectors. These vectors are called feature descriptors. A good descriptor should have the ability to handle intensity, rotation, scale and affine variations to some extent. One of the most famous descriptors is Scale-invariant feature transform (SIFT). SIFT converts each patch to 128-dimensional vector. After this step, each image is a collection of vectors of the same dimension (128 for SIFT), where the order of different vectors is of no importance.
  Codebook generation
  The final step for the BoW model is to convert vector-represented patches to "codewords" (analogous to words in text documents), which also produces a "codebook" (analogy to a word dictionary). A codeword can be considered as a representative of several similar patches. One simple method is performing k-means clustering over all the vectors. Codewords are then defined as the centers of the learned clusters. The number of the clusters is the codebook size (analogous to the size of the word dictionary).
  Thus, each patch in an image is mapped to a certain codeword through the clustering process and the image can be represented by the histogram of the codewords.

We construct a Bag of Words dictionary because the SIFT descriptor for each image has different dimensions. The Bag of Words dictionary helps us compute image descriptors that are of the same dimension which enables us to perform clustering the set of image descriptors.

- **Hierarchical Clustering** is a method of cluster analysis which seeks to build a hierarchy of clusters. We use the Agglomerative approach which is a "bottom up" approach. Each observation starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy.
  We perform hierarchical clustering on the image dataset and use the centroids thus obtained as reference points for the AP Clustering step. Since the size of the dataset is large, we cannot perform AP Clustering directly on it because of the limited configuration of regular computers.

- **AP Clustering** is a clustering algorithm based on the concept of "message passing" between data points. Unlike clustering algorithms such as *k*-means or *k*-medoids, affinity propagation does not require the number of clusters to be determined or estimated before running the algorithm. Similar to *k*-medoids, affinity propagation finds "exemplars", members of the input set that are representative of clusters.

Algorithm Steps

1. Obtain the set of bags of features.
   - Select a large set of images.
   - Extract the SIFT feature points of all the images in the set and obtain the SIFT descriptor for each feature point that is extracted from each image.
   - Cluster the set of feature descriptors for the amount of bags we defined and train the bags with clustered feature descriptors (we use the K-Means algorithm).
   - Obtain the visual dictionary.

2. Obtain the Bag of Words descriptor for given image.
   - Extract SIFT feature points of the given image.
   - Obtain SIFT descriptor for each feature point.
   - Match the feature descriptors with the dictionary we created in the first step
   - Build the histogram.

3. Obtain the points representing 1M images
   - Perform hierarchical clustering on the image dataset and obtain a fixed number of centroids.
   - These centroids will represent the image dataset.

4. Obtain the clusters in the dataset
   - The centroids after hierarchical clustering are given as input to AP clustering algorithm
   - The final number of clusters are obtained thereafter.

Program Design and Explanation

The different methods in the file are:

- currentTime is used to print the time on the console. It takes no input and returns the current time in string format
- plotCluster visualizes clustering performed on any numpy array. It takes as input the numpy array to be plot (number of observations, feature vector size), numpy array of its cluster centers (number of cluster centers,), numpy array containing label of each observation (number of observations,) and integer holding the value of number of centroids/ clusters.
- calculateSilhouette calculates the silhouette score after a clusterinf has been performed. It takes as input the numpy array clustering was perfomed on (number of observations, feature vector size), numpy array containing label of each observation (number of observations,) and a string containing the name of metric being used calculate distances between instances in the array being clustered. It returns the silhouette score as a float value.
- plotArray is used to plot any 2-D numpy array. It takes as input a numpy array (m, n) and prints the plot on the console.
- computeImageSIFT reads each file in the directory with the specified extension, converts it to grayscale, calculates its SIFT descriptor and adds it to a Bag of Words K means Trainer object. It takes as input the name of the directory images are in as a string and the extension of the images in the directory as a string and returns the Bag of Words Trainer object.
- computeBOWDictionary clusters a Bag of Words K means Trainer object and creates an image descriptor extractor object. It takes as input the Bag of Words K means Trainer object and returns the image descriptor extractor object.
- computeImageBOW reads each file in the directory with the specified extension, converts it to grayscale, calculates its SIFT descriptor, computes its image descriptor from the image descriptor object, appends the descriptor to a list and converts the list into a numpy array. It takes as input the name of the directory images are in as a string, the extension of the images in the directory as a string and an image descriptor extractor object and returns the computed descriptor matrix as a numpy array.
- performHierarchical performs hierarchical clustering on a numpy array and returns the numpy array of computed labels and an array containing centroids generated after the clustering. It takes as input a numpy array (number of observations, feature vector size), an integer value that holds the value of number of clusters, a string containing the name of metric being used calculate distances between instances in the array being clustered and a string containing the name of linkage type. It returns a numpy array containing label of each observation (number of observations,) and another numpy array of its cluster centers (number of cluster centers,).
- performAP performs AP Clustering on a numpy array and returns its attribute values. It takes as input a numpy array (number of observations, feature vector size), an integer holding the value of damping factor, an integer holding the value of number of convergence iterations and another integer holding the value of maximum number of iterations. The values being returned are numpy array of its cluster centers (number of cluster centers,), numpy array containing label of each observation (number of observations,) and integer holding the value of number of centroids/ clusters.
- main method calls all the other methods in a sequence to accomplish the project problem statement. The sequence of calls is computeImageSIFT, computeBOWDictionary, computeImageBOW, performHierarchical, performAP, plotArray, plotClusters and calculateSilhouette. Method currentTime is called to print on the console the time after each step is finished. The name of directory containing image files, the extension of image files and

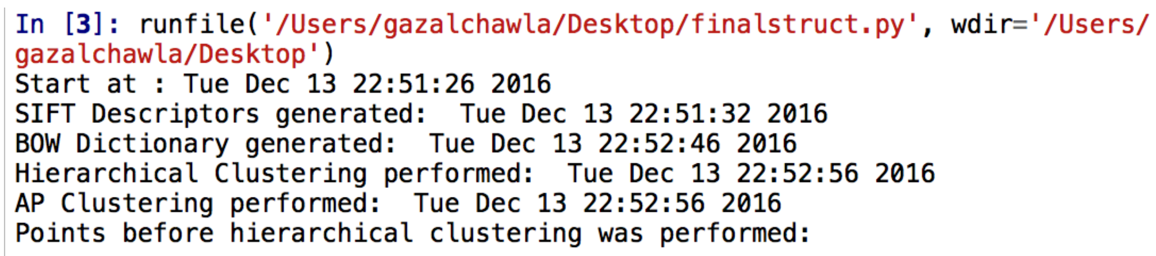
the number of clusters in the Bag of Words dictionary are set to their respective values in the main method.

Steps of Execution

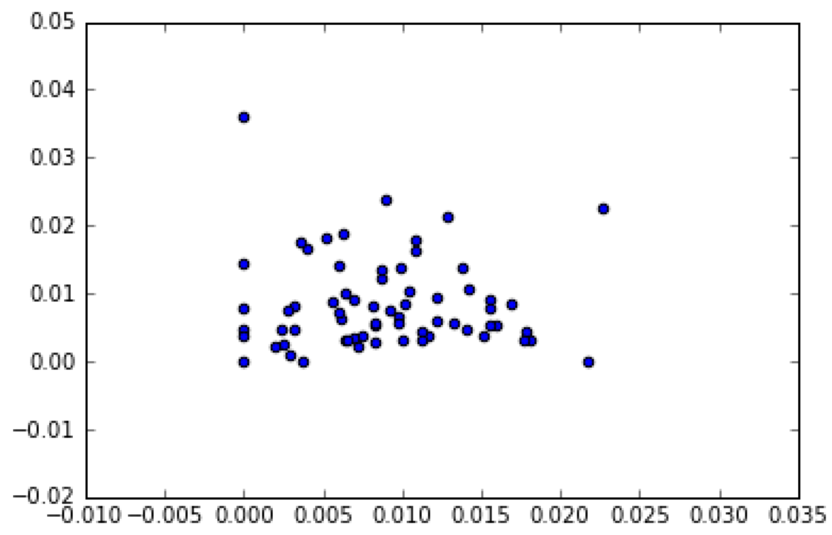
1. Download Anaconda with Python version 2.4 from https://www.continuum.io/downloads and install it on your machine.
2. Launch Spyder IDE
3. Install opencv3 by following the instructions on the link https://anaconda.org/menpo/opencv3 (Remember to run this command on either the Terminal(Mac) or Command Prompt(Windows) and not the IPython console on Spyder IDE)
4. Open final.py file in Spyder IDE, change the variable values for pathImageDir, extension and noCluster to your respective desktop and image dataset values.
5. Run the file in Spyder IDE on an IPython console.

Screenshots for one case

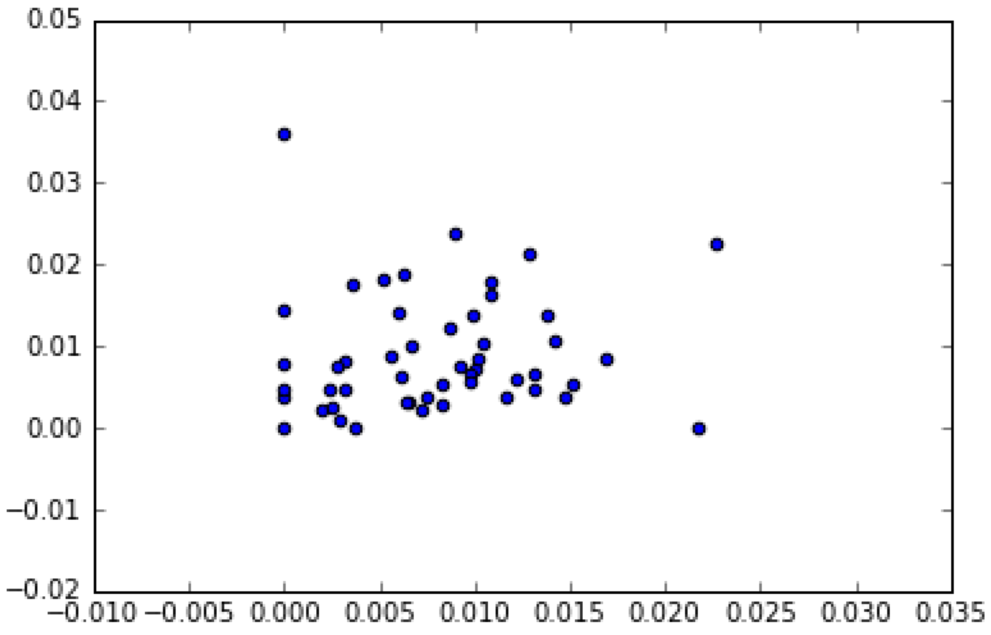
For dictionary size = 100, dataset size = 3000, number of centroids after hierarchical clustering = 50



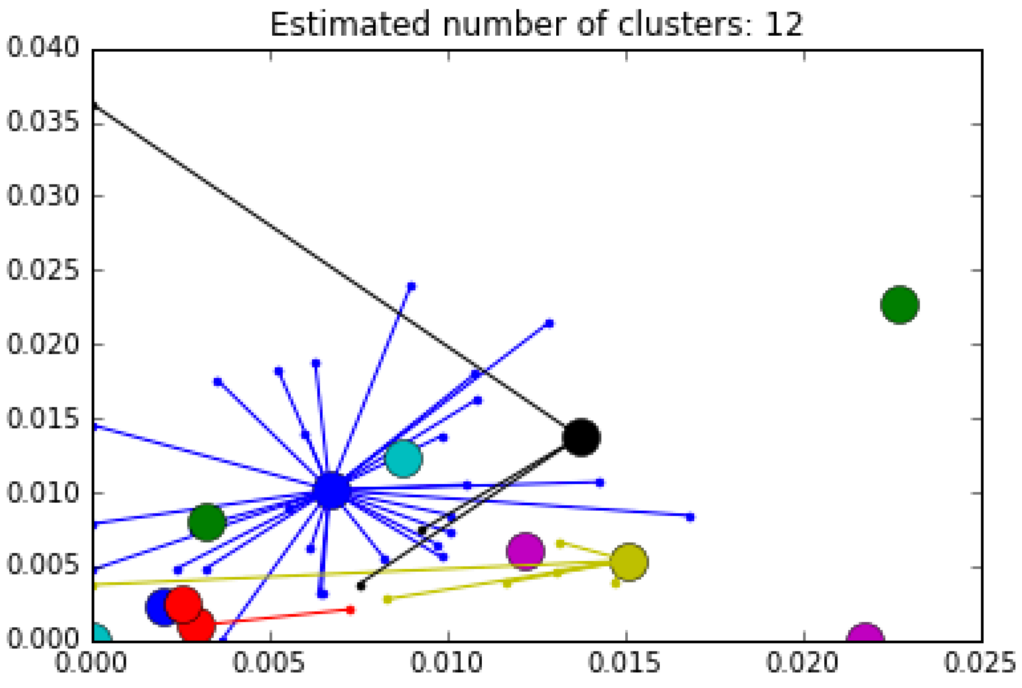
```
In [3]: runfile('/Users/gazalchawla/Desktop/finalstruct.py', wdir='/Users/
gazalchawla/Desktop')
Start at : Tue Dec 13 22:51:26 2016
SIFT Descriptors generated:  Tue Dec 13 22:51:32 2016
BOW Dictionary generated:  Tue Dec 13 22:52:46 2016
Hierarchical Clustering performed:  Tue Dec 13 22:52:56 2016
AP Clustering performed:  Tue Dec 13 22:52:56 2016
Points before hierarchical clustering was performed:
```



```
Points after hierarchical clustering was performed:
```



```
Points in final clusters after AP clustering was performed:
```



```
Silhouette Score for clustering:  0.225686297872
Ends at : Tue Dec 13 22:52:56 2016
```