MWDB PROJECT - PHASE 3

Contributors:

Ashwin Karthik Ambalavanan

Ajay Gunasekaran

Aditya Vikram Sharma

Balaji Gokulakrishnan

Srivathsan Baskaran

Akbar Jamal Abbas

Student- Arizona State University

The codebase in this phase is used for graph analysis, clustering, indexing and classification of images using features extracted from a set of images in flickr as given in the link: http://skuld.cs.umass.edu/traces/mmsys/2015/paper-5

Check the detailed description of tasks in the pdf file attached with the project. The task code SHOULD be run in the order described for correct execution.

* The "MWDB Project/Phase 3/Code" folder contains the entire project.
* The "MWDB Project/Phase 3/Outputs" folder contains document outputs for each individual task.
* The "MWDB Project/Phase 3/Report" describes in detail what has been done in the project.

Description of items MWDB Project/Phase 1/Code Folder:

1. Code/CSV- Contains required csv files for each task

2. Code/Tasks- Contains the Task code for each task.

3. Code/PHASE 3 SAMPLE INPUTS - Has the example inputs given to each task.

* Code/CSV/Img\_Img\_Graph\_k\_x - Img-Img similarity graphs for different values of k=x outputted by Task 1 (used for subsequent tasks).
* Code/CSV/Task x.html - HTML files used to visualize the output of various tasks in a webpage. They are outputted by every task in this phase.
* Code/CSV/Task 1- Contains the combined CSV files for models CM, CM3x3, CN, CN3x3,CSD,GLRLM,GLRLM3x3, HOG, LBP, LBP3x3 of all the locations.

The task code described below SHOULD be run in the order described for correct execution.

Code/Tasks/Task\_1.py- The code forms a image-image similarity graph for the images given in the dataset such that for each image there are 'k' outgoing edges to k-different images which are most similar to the image at hand. The similarity measure used is cosine similarity.

Caution: Code can take extremely long time to run because of running on large dataset and combining scores from all the models.

Code/Tasks/Image\_Visualize.py- Sample code to visualize output of tasks in webpage.

Code/Tasks/Task\_1.py- The code gives the implementation of creation of the Image-Image similarity graph with ‘k’ outgoing edges from each node. The resulting graph is to be used as inputs for subsequent tasks.

Code/Tasks/Task\_2.py- The code gives the implementation of two clustering algorithms, Nearest neighbours and Spectral Clustering to find ‘c’ clusters in the image-image similarity graph for a user given ‘c’.We visualize the resulting images in a web browser.

Code/Tasks/Task\_3.py- The code gives the implementation of the page rank algorithm to find the 'k' most dominant/significant/important images in the dataset as given by the page rank algorithm for a user given 'k'.

Code/Tasks/Task\_4.py- The code gives the implementation of the page rank algorithm to find the 'k' most relevent/significant/important images for a set of images and a user given 'k' as given by the page rank algorithm.

Code/Tasks/Task 5a.py (A): This code gives the implementation of LSH with Hamming distance as the similarity measure. This code returns an in-memory index structure containing the given set of vectors when given the values of K and L

Code/Tasks/Task 5b.py (B): The code implements similar image search using this index structure and a combined visual model (all colour models) and for a given image and t, it visualises the t most similar images (also outputs the numbers of a unique and overall number of images considered).

Code/Tasks/Task\_6a.py (A): This code uses the sample image/label pair as the input and assigns a label to rest of the images in the database using the k-nearest classifier algorithm. We visualize the labeled results in a web browser.

Code/Tasks/Task\_6b.py (B): This code uses the sample image/label pair as the input and assigns a label to rest of the images in the database using personalized page rank classifier algorithm. We visualize the labeled results in a web browser.