Readme - Duplicate Image Detection

#duplicate_detection.py

The file is written for the detection of duplicate images for a dataset.
It requires the following libraries to run:

os, glob, Image, hashlib, shutil, imagehash

I have included the imagehash library along with the code. Install this module for the program to run.

The outline of the code is as follows:

1. For Exact Duplicate Detection: I used an md5 hashmap for each image to generate a key for each image. While building a hash dictionary, I obtain the hashkey for each image, I check if that hashkey is already present in the hash dictionary. If yes, then an exact copy of this image has already been processed. So we include this in the duplicate list. If we don't find the hashkey in the dictionary then this is a different image and hence added to the dictionary along with its key.
2. For Near Duplicate Detection: I used a perceptual hash algorithm particularly dhash algorithm implemented in the imagehash module. This is very useful and helpful for nearly duplicate image detection. So, again a similar approach to that of above for obtaining the entire hash dictionary and duplicate list. Nearly duplicate images have the same dhash value and hence the near duplicates can be detected.

The above two are the most important parts of the code. Apart from the above two there are modules for copying all the duplicates and unique image files to different folders and writing the exact locations of the duplicates and unique images in different files. The code in the main module also allows to add further images to the dataset after the initial iteration.

Bottlenecks in the code:
However, it is to note that after each iteration, if the same destination folder is given, the previous files will be over written but the rest will remain untouched.
For ex:- for exact duplicates: duplicates- 1.jpg, 2.jpg, unique- 3.jpg, 4.jpg, 5.jpg (but 3 and 4 are nearly duplicate). Now if you give the same exact folder again while running for near duplicate detection: then Duplicates- 1.jpg, 2.jpg, 3.jpg unique- 3.jpg, 4.jpg, 5.jpg
The files in each of the folders are overwritten. However 3.jpg in the unique file remains untouched. So make sure to give different destination folders at every time or delete the old files and re-run the program.
Also the log file locations of duplicate and unique image files are always opened in append mode. So running the code on the same dataset multiple time will result in redundant entries in these log files.

I have tested the code on exactly similar images, near duplicate images, scale variation images. The code is found to be producing accurate results for all the above cases.

* Stage-1 (exact duplicate) ✓
* Stage-2 (very near duplicates) ✓

extended to

1. image converted from png to jpg ✓
2. image scaled ✓ (partial)

Thus I was able to code my approach which is able to identify very near duplicates for sure, but those with scaling, if there is a very large difference in the scaling then they are treated as different images and hence they are not very near duplicate according to my approach. But apart from that, it is robust to very near duplicate image detection.

Kindly get back to me if you have any further queries.

– Murali Raghu Babu.B

muraliraghubabu1994@gmail.com
IIT Guwahati