LAB 4: DUE 17 NOVEMBER 2016

In this assignment you will automatically "colorize" face images using a large data base of face images. The algorithm will proceed as follows. Given a grayscale query image, match that image to a large training set of color images. Transfer the color from matched image to the query image.

Task 1: Baseline Implementation (50 pts)

- Compute an image descriptor for the query face.
- Compute image descriptors for the training set of face images.
- Retrieve the training image that is most similar to the query.
- Transfer color from the matched training image to the query image.

Images in the testset directory are your grayscale query images. Images in the trainset directory are your large data base of color training images. Your task is to compute colorizations for each query image using the training data base.

For this task we will use Tiny Images (32x32 pixels images) as image descriptor and SSD as metric. For transferring the color, you can experiment with different color spaces. YCbCr, L*a*b*, HSV…

You can use libraries for the resizing and the color transformation to other color spaces.

The color transferring is similar to what we did in Lab1 or Lab2. Convert the image to a color space based on luminance and chrominance, use the query image as your luminance channel and the chrominance channels from the best match.

Task 2: Reduce dimensionality with PCA (50 pts)

In the Tiny Images paper, they use the first 19 principal components of the images, instead of 32x32 vectors, to speed up the comparison process. To compute the PCA, you have to create a large matrix with all your data (image descriptors) where each row is a flattened version of it.

This is a pretty large dataset, so the matrix will be big. Don't try with the whole data set before you get it working. Search for a decent implementation/library of PCA that can handle this amount of data. I tried sklearn in Python and it seems to handle it all right.
http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

Measure query times with between Task1 and Task2 and compare. Only the search time.

Notes:

- There are 30.000 images in the training set. A trivial implementation will be slow.
- Try for a few images and check that everything is working, before you run it for the whole dataset . Some of the images in the test set are in the training set, you can find one of them manually to check that you are getting the correct result.
- You can also compute the image descriptors of the training set once and save them to a file, so you don't have to recompute them every time.
- You compute PCA once for the training set, the you transform your data set and keep the number of principal components that you want as your descriptors. For your query, you simply use the same transformation for the image and use the resulting vector as your descriptor.

Extras.

- Experiment with different sizes of images and number of PCs, and different metrics, such as Normalised Cross-correlation.
- Experiment with different color transfer methods such as: https://github.com/jrosebr1/color_transfer