EE 610 Image Processing (July-Nov 2017)

Assignment 1 : Basic Image Editor

- Build a form (GUI) in with the following elements [7]:
  - Image display area
  - Image load button that opens a file selector that does not crash if no image is opened. Note, a grayscale image should be opened, not a color one. If the image is in color, first convert it to greyscale
  - Several image manipulation buttons that do the following to the image being displayed:
    - Equalize histogram
    - Gamma correct (ask for input gamma upon pressing the button)
    - Log transform
    - Gaussian blur (ask for width D_0; default 3.0)
    - Butterworth sharpening (ask for width D_0, order n, and amplification of the lowest frequency as a percent of amplification of the highest i.e. a in Figure 4.31 of the textbook; default 3.0, 2, 50% respectively)
    - Undo all changes (display originally loaded image)
  - Save current image button
  - It also has the following info buttons:
    - Display image
    - Display histogram
    - Display Fourier transform magnitude
    - Display Fourier transform phase
- Prepare a report in with the following sections and submit pdf [5]:
  - Approach used to build the software application
  - Main challenges faced
  - Results on greyscale version of some image that you find interesting.
    - Three sub-images (choose different parts of the image with different patterns)
    - Histograms of the three sub-images, with observations on why they are different
    - Histograms of histogram equalized versions of the three sub-images
    - Display of the histogram equalized sub-images
    - Display of the gamma corrected equalized sub-images with 0.5, with comments
    - Display of the gamma corrected equalized sub-images with 2.0, with comments
    - Display of the log-transformed sub-images with comments
    - Display of the log-transformed sub-images with comments
    - Display of the Gaussian blurred sub-images with comments
    - Display of the Butterworth sharpened sub-images with comments
    - Display of Fourier magnitudes of the original sub-images and that of their Butterworth sharpened version, with comments comparing the two
  - Appendix with code
    - Using coding best practices, e.g.:

- Proper indentation
- Informative variable names
- Comment about every single line of the code explaining the role of each variable, e.g.

```
## Initialize Co-occurrence matrices

poolFetSize = poolImSize*poolImSize*poolImDepth # Pooled feature
size is the product of pooled image squared and pool image
number of channels (depth)

fets2Keep = min(fets2Keep,poolFetSize) # Just in case we chose
too many features

coOcPos = np.zeros((fets2Keep,fets2Keep)) # Initialize positive
co-occurrence matrix
coOcNeg = np.zeros((fets2Keep,fets2Keep)) # Initialize negative
co-oc matrix


## Iterate through image pairs

for nIter in range(0,nBatches): # For all batches
    for nSample in range(0,nTrainBatchSize): # for all samples
in a batch
        img1Class = 100 # some large number
        while img1Class > 5: # We only want to train for the
first five
            img1Num = np.random.randint(nImages) # pick a random
image
            img1Class = clas[img1Num,0] # Find its class

        img2Class = 100 # Pick class 100
        while img2Class != img1Class: # Loop till the classes of
the two images are the same
            img2Num = np.random.randint(nImages) # Pick a random
second image
            img2Class = clas[img2Num,0] # Find its class
```

- Notes:
  - Bonus +1 for using Octave.
  - Bonus +3 for using Python.
  - You can look at code available on the Internet, but do not directly copy. At least, customize.
  - Comments in the code should be your own.
  - F grade for copying from another student.