scc0251/5830— Prof. Moacir Ponti

Assignment 2

Try to develop and implement yourself, without any help from your colleagues.

Image Enhancement

In this assignment you have to implement functions in order to enhance images using a series of operations.

Your program must have the following steps:

1. **Parameter input** (read from input):
   a) file name for the input image,
   b) γ parameter (to be used in gamma adjustment),
   c) a parameter (to be used in sharpening),
   d) b parameter (to be used in sharpening),
   e) a image show flag (0/1) which is 1 for showing the image windows and 0 otherwise.
2. **Read** the input image;
3. **Apply the transformation** of pixels following the operations described in the next section (log, gamma, histogram equalisation, sharpening);
4. (Image show flag == 1) **Show all images**: input image and all transformed images (if show flag is 1), waiting for the user to press some key;
5. (Image show flag == 1) **Show the histogram** of all images (only if show flag is 1), waiting for the user to press some key;
6. **Output** for each image: the Root Mean Squared Error comparing the input image with the processed images (see patterns for input/output below for details).

1 Enhancement operators

1.1 Logarithm

$$T(r) = c\log(1 + |r|),$$

where c is:

$$c = \frac{255}{\log(1 + R)},$$

and R is the maximum intensity value found in the image.

1.2 Gamma adjustment

$$T(r) = r^{\gamma},$$

1.3 Histogram equalisation

$$T(r) = \frac{(L-1)}{MN} hc(r),$$

where $hc(r)$ is the cumulated histogram of the input image (use your own histogram function as described below).

1.4 Sharpening

First, you have to produce an image $b(x, y)$ by computing the convolution of the input image $f(x, y)$ with the following filter:

$$w(x,y) = \begin{array}{c|c|c} 0.05 & 0.1 & 0.05 \\ \hline 0.1 & 0.4 & 0.1 \\ \hline 0.05 & 0.1 & 0.05 \end{array}$$

Therefore, you will produce:

$$b(x, y) = w(x, y) * f(x, y)$$

Then the processed image will be:

$$g(x, y) = a \cdot f(x, y) + b \cdot (b(x, y) - f(x, y))$$

Look at the function `imfilter` from OpenCV documentation in order to perform the convolution.

2 Histogram computation and display

You also have to show the histogram of the images. In order to do that you have to implement your own function (that may be used in the Histogram equalisation operator). However you may use some existing function in order to plot the histogram.

Functions for plotting graphs and histograms are available in the package `matplotlib`.

3 Root Mean Squared Deviation

You also have to show the Root Mean Squared Deviation from the input image to all processed images. The RMSD is defined as:

$$\epsilon = \sqrt{\frac{\sum_x \sum_y \left(f(x, y) - g(x, y)\right)^2}{n \cdot m}}, \tag{1}$$

where f is the input image, g is the processed image and $n \times m$ is the resolution of the images. Note that f and g must have equal resolution.

Show the RMSD rounded to 4 decimal places (use e.g. `print("%.4f" % e)`)

4 Patterns for input/output

Example of input (filename, gamma, a, b, flag):

```
arara.jpg
1.25
0.7
0.3
1
```

Example of output (RMSD for log, gamma, histogram-equalised and sharpening operations):

```
RMSD\n
L=64.1213\n
G=60.1031\n
H=23.4512\n
S=18.6192\n
```

Where `\n` is a line break. Note that the run.codes system will compare the output trying to find an exact match, so please follow the exact output pattern.

5 Functions

Your program **have to define** the following functions:

- `g = imLog(f`, to perform the logarithm enhancement, returns a processed image g
- `g = imGamma(f, y)`, for the gamma adjustment
- `g = imEqualHist(f)`, for the histogram equalisation
- `g = imSharp(f, y)`, for the Sharpening
- `h = imHistogram(f)`, returns a histogram in form of a vector of counts of each grey level
- `showHistogram(h)`, show in the screen a window with the histogram h

6 Instructions

- Individual homework assignment
- Deadline as found in the Run.Codes system

Submission Use the Run.Codes system for submission, including ONLY the `.py` file

It is **obligatory** to comment your code. As a header, use **name and USP number** (your grading will be discounted if this is missing).

Questions should be asked in the hours of the teaching assistant, or via email with the subject `[IP HA2] <question>`, especifying the question.

The source-codes will be tested for plagiarism. If any copy is detected, even in few parts of the code (such as a single function), the homework will be assigned with grade 0.