scc0251/5830— Prof. Moacir Ponti

Teaching Assistants: Gabriel Cavallari and Vinicius Costa

Assignment 1 : intensity transformations

Try to code the assignment by yourself. Plagiarism is not tolerated.

Image Intensity Transformation

In this assignment you have to implement image intensity transformations using different strategies. Read the instructions for each step. Use `python` with the `numpy` and `imageio` libraries.

Your program must allow the user to provide parameters in order to read the image and apply the transformation. The input image is grayscale and in the *uint*8 format.

1. **Parameter input**:
   a) filename for the reference image r
   b) the transformation T (1, 2, 3 or 4)
   c) parameter S to save the modified image (if $S = 1$)
   d) parameters c (integer) and d (integer) if $T = 2$
   e) parameters W (integer) and λ (float) if $T = 4$
2. **Compute the modified image** m, applying the transformation T to the image r.
3. **Compare** m, the image with modified intensities, with the reference image r.
4. **Print** in the screen the root squared error between m and r.
5. **Save** the modified image if $S = 1$.

Transformations

The functions are applied pixel-by-pixel. Here, $T(i)$ denotes the transformation T(x,y) applied to the location (x,y) of the image.

1. Inversion: $T(i) = 255 - i$;
2. Constrast modulation: $T(i) = (i-a)\dfrac{(d-c)}{(b-a)}+c$; This linear transformation modifies the range of the input image [a; b] into a new range [c; d]. That means: a is the lowest image intensity value, b the highest image intensity value; c and d are the parameters of the new lowest and highest values, respectively.
3. Logarithmic Function: $T(i) = 255\dfrac{\log_2(1+i)}{\log_2(1+R)}$, where R is the highest image intensity value;
4. Gamma adjustment: $T(i) = Wi^{\lambda}$

Parameters input

The code below shows how to read de inputs and how to save the modified image.

```
filename = str(input()).rstrip()
input_img = imageio.imread(filename)
method = int(input())
save = int(input())

if method == 1:
    ...

if method == 2:
    c = int(input())
    d = int(input())
    ...

if method == 3:
    ...

if method == 4:
    W = int(input())
    lambd = float(input())
    ...

if save == 1:
    imageio.imwrite('output_img.png',output_img)
```

Comparing with reference

Your program must compare the modified image m with the reference image r. This comparison must use the root squared error (RSE). Print this error in the screen, rounding to 4 decimal places.

$$RSE = \sqrt{\sum_i \sum_j (m(i,j) - r(i,j))^2}$$

Note this formula does not divide the error by the number of pixels. It is a modification of the Root Mean Squared Error, showing the sum of the errors in all pixels. When computing the error, use the images values as $float$ to avoid memory overflow.

Input/output examples

Input example: filename of reference image *airplane.png*, function 2, parameters: S = 1, $c = 50$, $d = 100$

```
airplane.png
2
1
50
100
```

Output example: RSE value in format `float`

```
15979.0811
```

Submission

Submit your source code using the Run.Codes (only the `.py` file)

1. **Comment your code**. Use a header with name, USP number, course code, year/semester and the title of the assignment. A penalty on the grading will be applied if your code is missing the header and comments.
2. **Organize your code in programming functions**. Use one function per type of transformation (1,2,3,4).