Color Project

This project requires you to implement and experiment with several programs that manipulate color spaces in images.

OpenCV support

The programs involve the following color spaces: sRGB, XYZ, Luv. OpenCV representation of these color spaces is different from what was discussed in class in the following way: all values are mapped to the $[0-255]$ range and rounded to the nearest integer. Here are the OpenCv conversion routines:

```
output_image = cvtColor(input_image, cv2.FLAG)
```

FLAG values that are relevant for us are:

COLOR_XYZ2BGR, COLOR_BGR2XYZ,
COLOR_Luv2BGR, COLOR_BGR2Luv.

The project also requires OpenCV code for histogram equalization. Here is how it is used:

```
output_gray_image = cv2.equalizeHist(input_gray_image)
```

Programs

You are ask to write a total of 6 programs. Each program takes a color image as input and produces a color image as output. A program operates on a specific color space, and uses 1 out of 3 possible algorithms:

- Linear stretching.
- OpenCV histogram equalization.
- The histogram equalization algorithm described in class.

Please use the following names for your programs:

luv_lscl.py	luv_histeq.py	luv_classhisteq.py
xyz_lscl.py	xyz_histeq.py	xyz_classhisteq.py

Each program stretches the illumination component of the image over a specified window. The illumination components are the **L** part of the **Luv** representation, and the **Y** part of the **XYZ** representation. In each case the illumination component can only have integer values in the $[0-255]$ range.
This is not exactly as was described in class. OpenCV makes it easier for us by taking care of the range. Another point of difference is that the histogram equalization of OpenCV is not exactly as the one discussed in class.

What you need to implement

Most of your code should use OpenCV routines. You need to implement the two routines that do **linear stretching** and **class histogram equalization**.

Input and output

All programs have the exact same arguments: an input image, window specification, and the name of the output image. The window is specified in terms of the normalized coordinates w_1, h_1, w_2, h_2, where the window upper left point is (w_1, h_1), and its lower right point is (w_2, h_2). For example, $w_1 = 0$, $h_1 = 0$, $w_2 = 1$, $h_2 = 1$ is the entire image, and $w_1 = 0.3$, $h_1 = 0.3$, $w_2 = 0.7$, $h_2 = 0.7$ is a window in the center of the image. The provided example program shows how to read the arguments and go over the pixels of the specified window.

Evaluation

Your grade will depend on the following two components: correctness of your programs and your report of results of experiments. You are asked to experiment with your programs and report the following:

1. Describe strange behavior when colors appear to be changing. If this occurs it is an indication that the OpenCv code does not handle out of range values properly. You should report it and show an image where this occurs.

2. Among the 2 lscl programs decide which one is the best and which one is the worst. Show images that support your conclusion.

3. Among the 4 histeq programs decide which one is the best and which one is the worst. Show images that support your conclusion.

All experiments should be done with natural images and a window size of at least 50×50.

What you need to submit

- Submit the source code of all your programs.
- Submit a report of your experiments.
- Submit at least 3 and at most 10 images to justify the conclusions in your report.

Please notice that it is your responsibility to provide us with all the above information. You must be present when your project is being evaluated.

Due Date: TBA