CS2302 - Data Structures

Summer 2020

Lab 1

Due Friday, June 19, 2020

A gray-level (or black and white) image with r rows and c columns is represented as a $(r \times c)$ 2D array I, where $I[i, j]$ represents the intensity of pixel (i, j). $I[i, j] = 0$ represents a black pixel and $I[i, j] = 1$ represents a white pixel, while values in between represent different shades of gray. Similarly, a color image is represented as a 3D array of dimensions $(r \times c \times 3)$, where $I[i, j, 0], I[i, j, 1]$ and $I[i, j, 2]$ contain the red, green and blue intensities, respectively, of pixel at row i and column j.

For this lab you will practice with 2D and 3D arrays and slicing in Python and gain an understanding of the relative efficiency of both ways of dealing with arrays. Your task is to implement each of the following operations with two different functions, one that uses loops and only accesses individual array elements (as we do in Java) and another that takes advantage of slicing and operations on complete arrays as provided by Python.

1. Return an upside down version of the original image.

2. Return a mirrored version of the original image.

3. Return a version of the original image with the red and blue channels swapped.

4. Return an image with half the resolution of the original image. This should be done by creating an array that contains only the even-numbered rows and columns of the original array.

5. Return a gray level version of the original image. This should be done by computing a weighted average of the red, green and blue channels in the color image, using weights $\langle 0.3, 0.59, 0.11 \rangle$, which have been shown to produce more natural images than simple average.

6. Return the negative of a gray level image.

7. Return the binary version of a gray level image. A pixel in the binary image is set to 1 if the corresponding pixel in the original image is greater than 0.5, otherwise it's set to 0.

8. Return an image representing the vertical edges in a gray level image. A vertical edge at position r, c is the difference between pixel r, c and its neighbor to the right (E[r,c] = abs(I[r,c] - I[r,c+1]), where E is the edge image and I is the gray level image. Notice that E must have the same number of rows as I but one fewer column, since the pixels in the last column of I don't have a right neighbor.

9. Return an image representing the horizontal edges in a gray level image. Similarly to the previous item, a horizontal edge at position r, c is the difference between pixel r, c and its neighbor below (E[r,c] = abs(I[r,c] - I[r+1,c]) Notice that E must have the same number of columns as I but one fewer row.

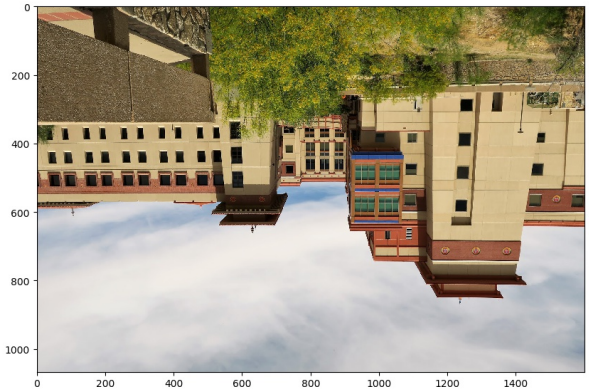
The starter program includes code to read and display an image array and to build a copy of the array using for loops and using slicing. Use the implementations of the copy operation as model for all the requested functions. Feel free to use your own images for your experiments.

Write a report describing your work, as explained in the syllabus. Use tables to show the running times for each operation and implementation, as well as comparisons between implementations of the same operation.

Appendix: Sample results

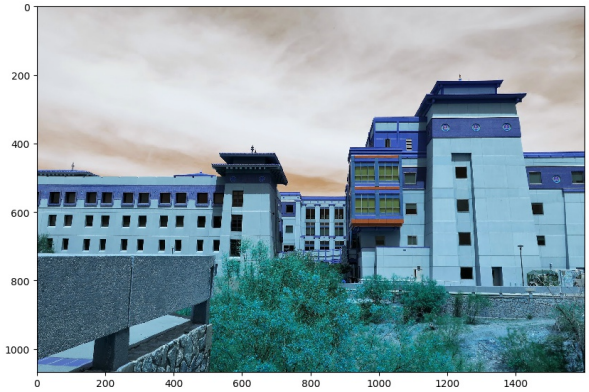
original

upside down



mirror

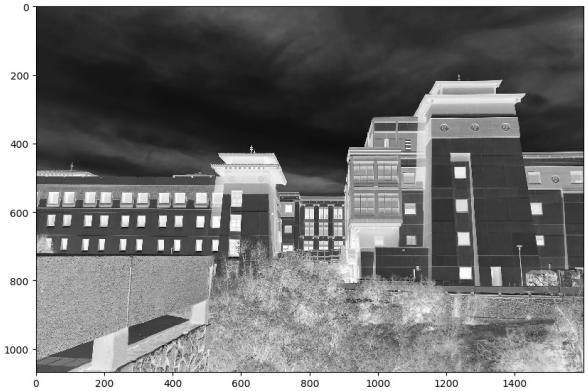
swap red and blue



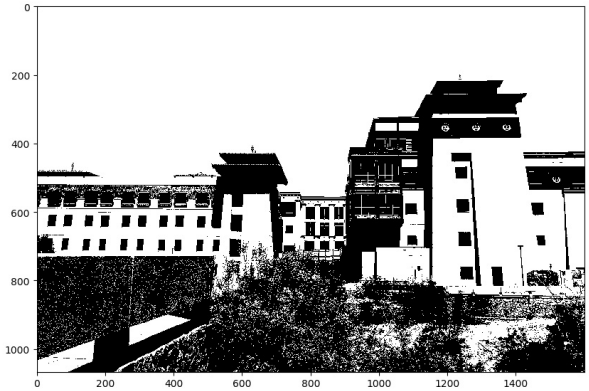
gray level

half resolution

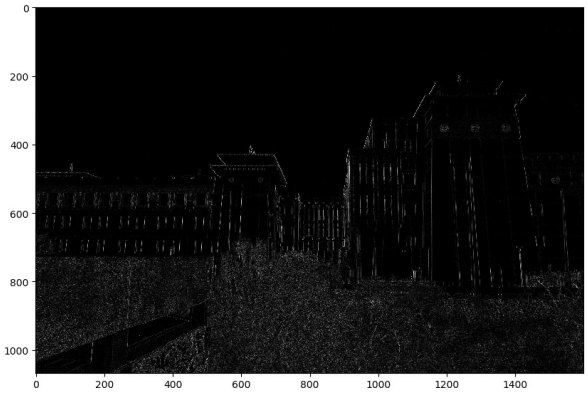
negative gray level



binary



vertical edges



horizontal edges