CSCE100 Introduction to Informatics
Fall 2023

Programming Assignment 3:
Computing Image-Based Statistics & Adaptively Modifying Images

Points: 100 points. Assignment Date: September 21, 2023 Due Date: September 28, 2023

Objectives

1. To familiarize with writing and running Python programs and the Python environment
2. To familiarize with the use of loops (e.g., for and while loops)
3. To familiarize with data structures, particularly arrays/lists
4. To familiarize with file input/output in Python
5. To be exposed to the use of built-in functions
6. To be exposed to the use of built-in modules or packages (e.g., import math)
7. To familiarize with the use of online documentations on Python

Relevance to Informatics or Data Science

1. A user interface program for reading and generating imagery data
2. A program for pre-processing or processing imagery data
3. A solution for computing basic statistics

Problem

Download from the Canvas site the zipped folder of a set of images[1]. Each of these images is formatted in the PGM format. These images will serve as input files to your program for you to experiment and test your code.

Download also the starter code. The starter code reads in the image file and writes the image array out to an output image file. Review the starter code to find out what the output filename is.

Now, given the starter code. First, print out a message to welcome user to the program. Second, prompt the user for an input filename. After this, the code will read in the input filename and store the image in the variable "raster", which is a 2-dimensional array. After that, you have two tasks to design and implement.

---

[1] These images are part of the contour image database. It is made available by C. Grigorescu, N. Petkov, and M. A. Westenberg (2003). Contour detection based on non-classical receptive field inhibition, *IEEE Transactions on Image Processing*, **12**(7):729-739.

- Task 1: Process the 2-dimensional array to generate four statistics: (a) average value (i.e., intensity) of the image, (b) the lowest intensity value of the image, (c) the highest intensity value of the image, and (d) the standard deviation of the intensity values of the image. Print out the statistics to the screen. See Appendix for the equation for computing the standard deviation of a series of numbers.
- Task 2: Modifies the image using some of the aforementioned statistics and explains the purpose of the modification (what effects does the modification have on the image) (*Hint*: Use your imagination and creativity on this! Come up with a way of modifying an image based on the statistics of the image itself – This leads to an adaptive, data-centric solution!)

And then, the code will proceed to write the updated "raster" to an output file.
Finally, print out a message to exit the program.

- The program extension part is required to display an explanation of the program (e.g., its expected range of input values) in the beginning before prompting the user for a search key. (5 points)
- The program is required to use loops to compute the four statistics. (5 points)
- The program is required to use loops to modify the image (5 points)
- The program is required to have documentation in the program that explains the purpose of the modification (10 points)
- The program is required to display an exit message (5 points)
- You must document your program (see https://devguide.python.org/documenting/).
    - Name, Date, Affiliation, a description of the program, what inputs does it need, what outputs does it generate (5 points)
    - Inline comments in the program (5 points)

Example Input/Output: None

```
Welcome to the Image Statistical Analysis and Processing program.
Please enter an input filename (without the .pgm): lions
number of rows: 512
number of columns: 512
average intensity: 89.41801452636719
lowest intensity: 4
highest intensity: 245
stdev of intensity: 51.35978447306207
Thank you for using the Image Statistical Analysis and Processing program.
```

The input image is (a) lions.pgm, and the output image is (b) lionsout.pgm:

(a) (b)

Hint: Can you tell how lionsout.pgm is generated? What is the effect achieved in lionsout.pgm?

Handin

1. The submission deadline for all handins is 11:00 AM September 27, 2022. **Late handins will *not* be accepted or graded.**
2. You are required to handin a screen capture of your "testing session" using your program. (5 points)
3. You are required to handin all program files. (5 points)
4. You are required to handin all output files generated from your testing session. (5 points)
5. You are required to handin online the above files to Canvas under Programming Assignment #3.

Think About

There are many different image formats. PGM is just one of many. GIF, TIFF, PNG, etc. are some that also accommodate color images. In such data files, an image is three dimensional: number of rows, number of columns, and number of channels. The number of channels represents, for example, RGB: red, green, and blue. When processing such images, one would need to have loops like this:

```
for x in range(0,numRows):
    for y in range(0,numCols):
        for z in range(0,numChannels):
            image[x][y][z] = image[x][y][z] + 5
```

If you are to process a large dataset of images to group the images into dark and bright images, how would you do it? If you are to process a large dataset of images to find all the images with high contrast, how would you do it?

Appendix (from Wikipedia: https://en.wikipedia.org/wiki/Standard_deviation)

Uncorrected sample standard deviation [edit]

The formula for the *population* standard deviation (of a finite population) can be applied to the sample, using the size of the sample as the size of the population (though the actual population size from which the sample is drawn may be much larger). This estimator, denoted by s_N, is known as the *uncorrected sample standard deviation*, or sometimes the *standard deviation of the sample* (considered as the entire population), and is defined as follows:[6]

$$s_N = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2},$$

where $\{x_1, \, x_2, \, \ldots, \, x_N\}$ are the observed values of the sample items, and $\bar{x}$ is the mean value of these observations, while the denominator N stands for the size of the sample: this is the square root of the sample variance, which is the average of the squared deviations about the sample mean.

This is a consistent estimator (it converges in probability to the population value as the number of samples goes to infinity), and is the maximum-likelihood estimate when the population is normally distributed.[*citation needed*] However, this is a biased estimator, as the estimates are generally too low. The bias decreases as sample size grows, dropping off as $1/N$, and thus is most significant for small or moderate sample sizes; for $N > 75$ the bias is below 1%. Thus for very large sample sizes, the uncorrected sample standard deviation is generally acceptable. This estimator also has a uniformly smaller mean squared error than the corrected sample standard deviation.