CS 4475 - Computational Photography Summer 2024

Individual Homework #1 - Due Friday May 24th before 8pm

Part 1: Installing Python, OpenCV, Numpy & Scipy.

You need to install Python (version 3.10.x) along with the Python OpenCV libraries (use OpenCV version 4.5.4) and all associated dependencies (such as numpy and scipy). We recommend you use anaconda and our enviornment .yaml file to easily install all dependencies at once. (See webpage for directions and download links.)

If you are having trouble with the installation, please post any errors / questions you have on EdDiscussions or contact us. Due to the diverse set of abilities in this, and the diversity of computing platforms out there, we expect that this will be a breeze for some, but harder for others. Please help each other setup and get started on this assignment. The goal of this assignment is to get you all setup so we can do more in later weeks, as well as get you up to speed with Python and image editing (if you are rusty). Yes, for some this would be trivial, those folks should step up to help others.

Part 2: Assignment Deliverables

The second part of the assignment requires you to use the python environment that you have created in Part 1. We want you to program five imaging functions. You may download the assignment files (zip file) that include the python template and a basic test file that will verify your functions output the correct type after you have written them (Note: simply because the tests succeed does not mean your function is doing the correct thing, you must verify this yourself, the test serves as a very basic check). Download the assignment files from Canvas or the course website.

This zipped folder contains the following files:

openCVImages.py - This is where you will write the code for the five functions we provide.

openCVImages_test.py - This can be used to test if your code is outputting the correct data type. (Note that it DOES NOT fully test your functions! Just because your code passes the test cases we provide does not mean that they are functioning correctly. You are welcome to enhance the test cases to test for functional correctness!)

README.txt - Some basic instructions on how to run the unit test and how to read and write images.

blueSky.jpg - A basic test image.

blueSky_2.jpg - Another test image. This is the same as blueSky.jpg but horizontally flipped -- you may use this image to get a sense of whether your flipHorizontal function is functioning correctly.

The function documentation details exactly what you need to do, but a basic description is provided here:

1 - numberOfPixels: This function takes in a grayscale image, and you must output an integer which represents the number of pixels in this grayscale image.

2 - averagePixel: This function takes in a grayscale image and outputs the average pixel value of that image as an integer.

3 - convertToBlackAndWhite: This function converts a grayscale image to a 1-bit black and white image. Details are provided in the function description.

4 - averageTwoImages: This function requires you to average two input images into the output. Further details are provided in the function description.

5 - flipHorizontal: This function flips an image across the vertical axis (mirroring columns from the left and right side). Further details are provided in the function description.

6 - tintRed: This function takes in a color image and sets the red channel to full on (255).

7 - yourFunction: You will need to make a function that takes in an image (color or grayscale, your choice) and does something cool to it.

Please turn in the following files:

openCVImages.py

Please put your name and GTID at the top of this assignment.

openCVImages.pdf

In the PDF we want you to write how you approached each of the seven functions above (a brief description). Also include (inside the PDF document, you do not need to upload them separately, although you may if you want) input / output images (of your choosing) for functions 3, 4, and 7. Please include your name in this PDF.

Grading Rubric:

1. numberOfPixels – 5 pts
   - number of pixels as integer is correct
2. averagePixel – 5 pts
   - average pixel value as integer is correct
3. convertToBlackAndWhite – 15 pts
   - image contains only values of 255 and 0
   - correctly thresholds to black and white image
4. averageTwoImages – 15 pts
   - each pixel is correct average
5. flipHorizontal – 20 pts
   - image is flipped horizontally (“mirrored”)
6. tintRed – 15 pts
   - each pixel in red channel is set to 255
7. yourFunction – 25 pts
   - graded based on attempted difficulty and/or creativity. (It doesn't have to be difficult if it is creative, and it doesn't have to be creative if it's difficult.)
     - either go for a cool looking picture or go for a new function/ task altogether