CS 470: Assignment 01

Programming Assignments (95%)

Your program MUST be written in **Python**.

Your code file should be named **A01.py**.

ALL submitted file(s) should be submitted as a zip/tar.gz file with the filename the same as your lowercase SITNET ID (e.g., realemj.zip)

The goal of this assignment is to perform **intensity slicing**.

You may NOT use OpenCV functionality for intensity slicing! However, you may use image loading, image saving, and grayscale conversion; you may also use image window functions for debugging. ***If you are not sure whether a library function is permitted, ASK ME!***

Your program should do the following:

- You will need to import the following:
    - import cv2
    - import sys
    - import numpy as np
    - from pathlib import Path
- Define a function **slice_image(image, lower_slice, upper_slice)**:
    - Copy the image (np.copy) → output.
    - Use np.where to check if output is ***less than lower_slice;*** if yes, use zero; otherwise, use the original pixel value.
    - Use np.where to check if output is ***greater than upper_slice;*** if yes, use zero; otherwise, use the original pixel value.
    - Return output.
- Define a function **main()**:
    - If the number of command line parameters is less than 5, print an error message and exit.
        - **sys.argv** is the list of command line parameters in Python.
        - Note that sys.argv[0] is the name of the program.
        - To get the total number of parameters passed in: len(sys.argv)
        - Note that the elements are strings, so for numbers you will need to convert:
            - E.g., x = int(s)
    - You should read in the parameters ***from the command line***:
        - File path to the image to load
        - The lower bound of the slice (convert to int)
        - The upper bound of the slice (convert to int)

        - The output directory to save images in
        - *Example*: python A01.py ./images/test.jpg 100 200 ./output
    - Load the image using cv2.imread() ***as a grayscale image***
        - You can either use the cv2.IMREAD_GRAYSCALE flag when loading OR just convert the image after the fact with cv2.cvtColor().
    - If the image is None, print an error message and exit.
    - Call slice_image() to generate the output image.
    - The output image filename will be:
        - out_filename =
          "OUT_"
          + Path(filename).stem
          + "_" + str(lower_slice)
          + "_" + str(upper_slice)
          + ".png"
    - The full output path will be
      (the output directory from the command line) + "/" + (output image filename)
    - Save the output image to the full output path using cv2.imwrite()
- Set up main() as the Python main function:
    - if __name__ == "__main__": main()

Testing Screenshot (5%)

I have provided several files for testing:

- Test_A01.py – the test program
- images/
    - test.jpg
    - other.jpg
- ground/
    - (Several output images to test against)

Copy these files/folders into the SAME directory as your python program.

You should be able to run the Test_A01.py directly.

You MUST run the tests and send a screenshot of the test results! Even if your program(s) do not pass all the tests, you MUST send this screenshot!

This screenshot should show clearly the final result of the test run ("OK" for all passing, "FAILED (failures=N)" for some or all failing).

Grading

Your OVERALL assignment grade is weighted as follows:

- 5% - Testing results screenshot
- 95% - Programming assignments

For the **PROGRAMMING** portion of the assignment:

Issue	***Penalty (in %)***
Code that has syntax errors	60
Incorrectly implemented slicing function	50
Incorrectly checking for the command line arguments	5
Incorrect reading of command line arguments	5
Incorrect loading of image	5
Not calling slice_image() in the main() function	5
Incorrectly saving the output image	5
Saved filenames incorrect	5
Functions named incorrectly	5
Files named incorrectly	5
Files not submitted as zip/tar.gz	5
Using forbidden library functions	60
Other stuff submitted (output images, etc.) BUT NO CODE SUBMITTED	100
Nothing submitted at all	100

In addition, I reserve the right to take points off for not meeting the specifications in this assignment description.

In general, these are things that will be penalized:

- **Code that is not syntactically correct**
- Sloppy or poor coding style
- Bad coding design principles
- Code that crashes, does not run, or takes a VERY long time to complete
- Using code from ANY source other than the course materials
- Collaboration on code of ANY kind; this is an INDIVIDUAL PROJECT
- Sharing code with other people in this class or using code from this or any other related class
- Output that is incorrect
- Algorithms/implementations that are incorrect
- Submitting improper files
- Failing to submit ALL required files