CISC/CMPE 472 – Assignment 1 (15%)

Image Processing

Due: February 3rd, 2023

For this assignment you are expected to implement all functions from scratch (with the exception of questions 1 and 2 only). Each question describes the structure of the function your code should implement. You may implement as many additional helper functions as required to keep your code legible. The use of image processing libraries such as OpenCV, SimpleITK, matplotlib, etc are not permitted unless explicitly stated in the question (numpy is permitted for basic algebraic matrix operations). Your code must be compatible with Python 3 (preferably a recent version). Code that does not run will not be graded. Your submission should be uploaded to OnQ as a zip file containing one python file containing your code named **CISC472_A1.py** and one PDF file containing your written responses. Ensure that your name and student number are in the header of each file.

1) **Read image** *[2 marks]*
   Write a simple function to read an image from a file. Your function should take in the path to the image you wish to read and return the image as an array. Your function must identify if the image is greyscale or colored. If the image is grayscale, your function should return a 1-channel image (e.g. shape = (height, width, 1)), for color images it must return a 3-channel image (shape = (height, width, 3)). You may use image processing libraries, such as OpenCV or matplotlib for this question.

2) **Display image** *[2 marks]*
   Write a simple function to display an image. Your function should take in an image as an array. The function should not return anything. You may use image processing libraries, such as OpenCV or matplotlib for this question.

3) **Linear filters** *[8 marks]*
   a) Implement a function that takes in a square kernel filter and an image as arguments and returns the result of applying the filter to the image. The filter may be of any size but will always be square.
   b) Test your function using a greyscale image of your choosing using the smoothing, edge detection and sharpening filters described in the notes. Include the code for testing your function in your python file. In your pdf include copies of the original image, as well as the result of applying each of the filters to your image (4 images total). Describe your observations.

4) **Non-Linear filters** *[8 marks]*
   a) Implement a function to apply mean filtering to an image. Your function should take in an image and the size of the filter as arguments. The function should return the result as an array.
   b) Using the same image as question 3, test your function. Include the code for testing your function in your python file. In your pdf include copies of the original image and the result of

applying the filter. Describe what you notice about the filtered image compared to the original image.

5) Depth modification *[8 marks]*

a) Write a function to change the depth of an image using a simple linear function. Your function should take as input the image to be modified and the desired depth and return the modified image. (Hint: pixel values should be rounded to the nearest integer)

b) Using the same image from questions 3 and 4, modify the depth of the image to that of a typical CT image (see lecture notes for value). Include this image in your PDF file and describe the differences you notice between the modified image and the original.

c) Reduce the depth of your original image to 4 bits. What changes? Reduce the depth of the original image to 2 bits. Include both images in your PDF along with your written observations.

6) Contrast enhancement *[9 marks]*

a) Write a function that will implement a simple contrast enhancement method. Your function should take as input the image to be modified, as well as 2 values that identify the range of values to be enhanced. Your contrast enhancement function should be this modified version of the one shown in lecture where *d* is the maximum depth of the image:

$$g(m,n) = \begin{cases} 0 & if\ f(m,n) < L_1 \\ d * \dfrac{f(m,n) - L_1}{L_2 - L_1} & if\ L_1 \leq f(m,n) < L_2 \\ d & if\ f(m,n) > L_2 \end{cases}$$

b) Test your function using the same image in previous questions with each the following ranges:

- $L_1 = 0.25 * d,\ L_2 = 0.75 * d$
- $L_1 = 0.45 * d,\ L_2 = 0.55 * d$
- $L_1 = 0.10 * d,\ L_2 = 0.60 * d$
- $L_1 = 0.50 * d,\ L_2 = 0.80 * d$

Record the modified images in your PDF and describe your observations of the changes you see in the image as the ranges are modified.

7) Generate a simulated image *[13 marks]*

a) Write a function that will generate random simulated greyscale images. The images should have a black background and contain a white ellipse (Figure 1). The location and dimensions of the ellipse should be randomly generated so that each time the function is called; it will produce a new image. The edges of the ellipse may not be cut off by the boundaries of the image. The function should take as parameters: The desired image dimensions, and the image depth in bits. The function should return the simulated image as an array. (Hint: for future assignments, it may help to implement this as 2 functions)

b) Generate a random image using the function from part a). Using your function from question 4, show the result of applying a mean filter with size 3, 7, and 15. Describe your observations as the kernel size increases.

c) Generate a random image using the function from part a). Using your function from question 4, show the result of applying a mean filter with size 5 once to your image. Next try repeatedly

applying this filter to the image 10 times, and 100 times. Describe your observations as the number of times that the filter is applied changes and include the resulting images in your PDF.

Mark breakdown

The mark breakdown for this assignment is shown in the table below. Please note that in addition to marks associated with each question, there are 11 marks dedicated for proper coding style. Code that is illegible or difficult to interpret or does not meet the style guidelines outlined below will be docked marks.

Question	**Marks associated**
1.	2
2.	2
3. a) b)	5 3
4. a) b)	5 3
5. a) b)	5 3
6. a) b)	5 4
7. a) b) c)	7 3 3
Coding style	10
Total	**60**

Style guidelines

Code for this assignment must be implemented in a single python file named **CISC472_A1.py.** Your code must contain an initial header that includes your name and student number. All import statements should be properly organized at the top of the file. All code must be properly structured within functions and all functions should be documented with initial comments describing the behavior of the function as well as the parameters and returns. The use of inline comments should be minimized and used only where necessary. Function and variables should have meaningful names that indicate their purpose (e.g. no single letter variable names). As mentioned earlier you must implement the functions as described, though you may use as many helper functions as you need to ensure that your code is clear and legible. For this assignment the use of image processing libraries such as OpenCV, SimpleITK, matplotlib, etc are not permitted unless explicitly stated in the question (numpy is permitted for basic matrix operations).

Reminder to make sure that your code runs using Python 3. If a function is causing errors that you are unable to fix before the deadline, please comment out that section of code. You may describe what you were trying to do in your PDF file to receive partial marks. Code that does not run will not be graded.