24-678: Computer Vision for Engineers

Carnegie Mellon University

PS2

Due: **9/23/2022 (Fri) 5:00PM @ Gradescope**

Issued: **9/14/2022 (Wed)**

Weight: **5% of total grade**

Note:

PS2-1 Converting a grayscale image to a pseudo-color image

In this problem, you will write a program that takes as input a grayscale image and converts it to a pseudo-color image. This process, called "pseudocoloring," assigns different colors to each greyscale level. There are various approaches to assigning colors to a greyscale image.

Figure 1 shows the most familiar pseudocoloring to mechanical engineers: a color plot of von Mises stress in structural analysis (left) and a color plot of pressure in fluid dynamics analysis (right). Since von Mises stress and pressure are both scalar values, the grayscale image would be sufficient to show the analysis results. It is, however, a common practice to display the result using a type of pseudocoloring scheme that shows a high stress/pressure value in red, a medium stress/pressure value in green, and a low stress/pressure value in blue.

Pseudocoloring is also commonly used for displaying a thermal image. Figure 2 shows a temperature measurement result shown in various types of color assignment, or color pallets. The most common ones used for commercial thermal cameras are shown in the top-middle and the bottom-middle.

Another common application of pseudocoloring is medical imaging. Figure 3 shows an original grayscale image from an X-ray camera (left), a pseudocolor image created by using an RGB color pallet (middle), and a pseudocolor image with an enhanced color pallet (right).

Your program should perform the following five steps to convert a grayscale image:

(1) Ask the user for an input grayscale image and display the input image in the first window.

(2) Find the lowest pixel value and the highest pixel value in the grayscale image.

(3) Make a look-up table to convert the lowest gray value to blue and the highest gray value to red. The other gray values should be mapped to rainbow colors by the method explained in the lecture.

(4) Using OpenCV functions, draw a cross in a circle to indicate the pixel of the highest gray value. Draw the cross and circle with white. If multiple pixels share the same highest gray value, place the cross and circle at the center of gravity of these pixels. Figure 4 shows a sample input image and output image.

(5) Save the final color image as input-filename-color.png and display the file in the second window.

Note: In the second and third steps, you must write your own Python code rather than using OpenCV built-in functions. In the fourth step, use OpenCV functions such as cv2.line() and cv2.circle().

Apply your program to the five grayscale images shown in Figure 5.

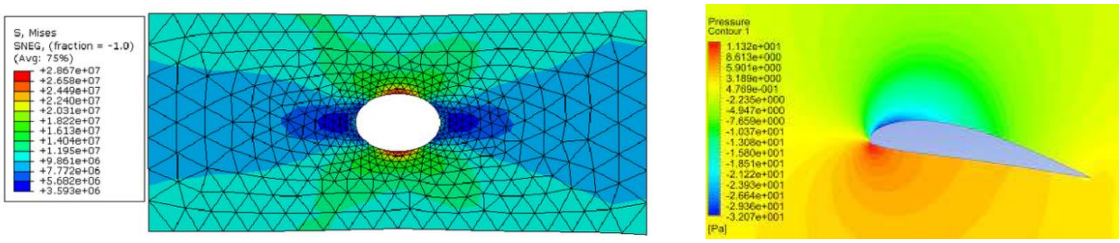

Figure 1. FEM analysis results are shown with pseudocoloring.
(left: von Mises stress, right: pressure)

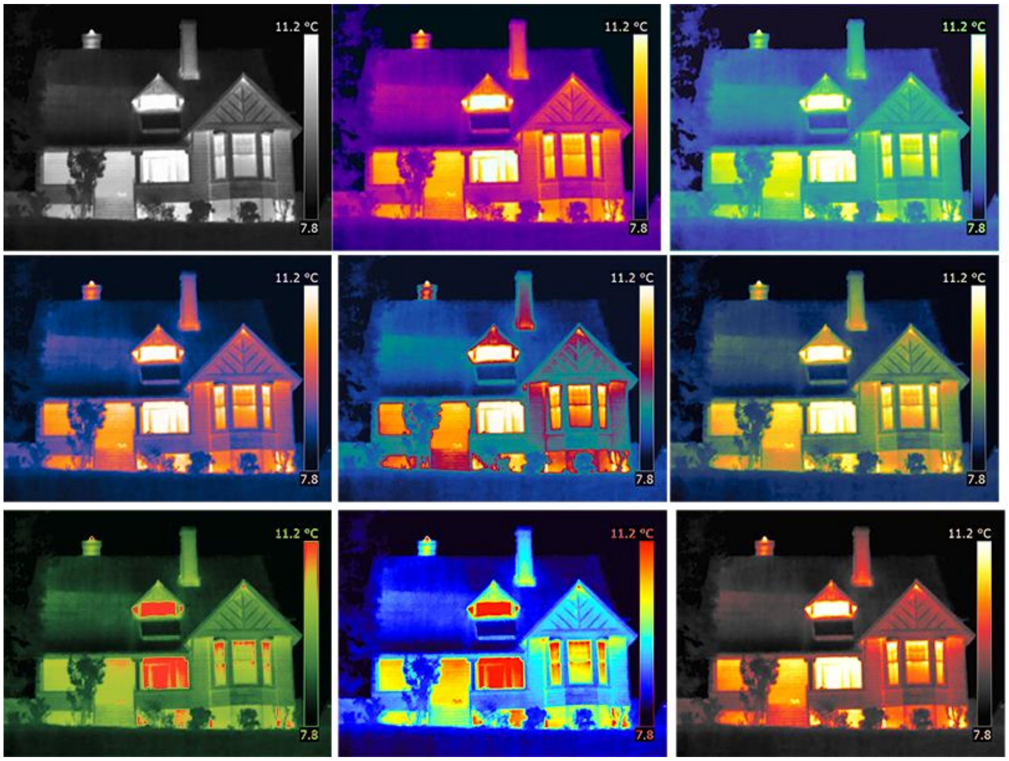

Figure 2. Thermal image of a house

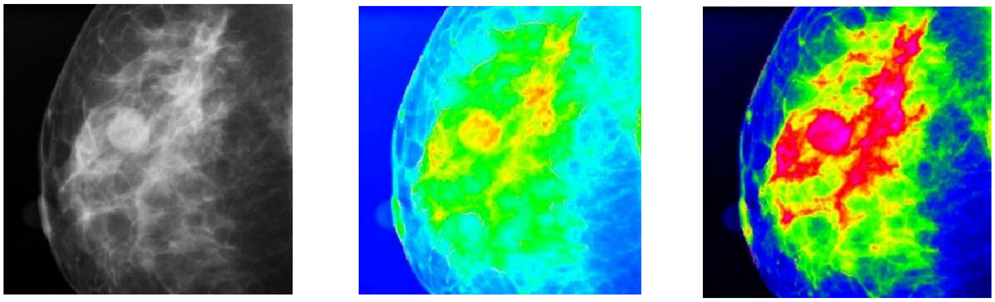

Figure 3. X-ray image for breast cancer diagnosis

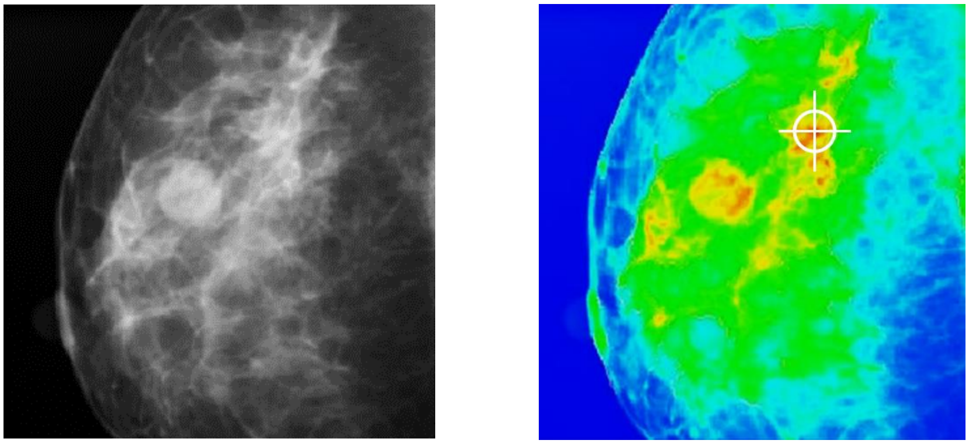

Figure 4. Input (left) and output (right) of your program

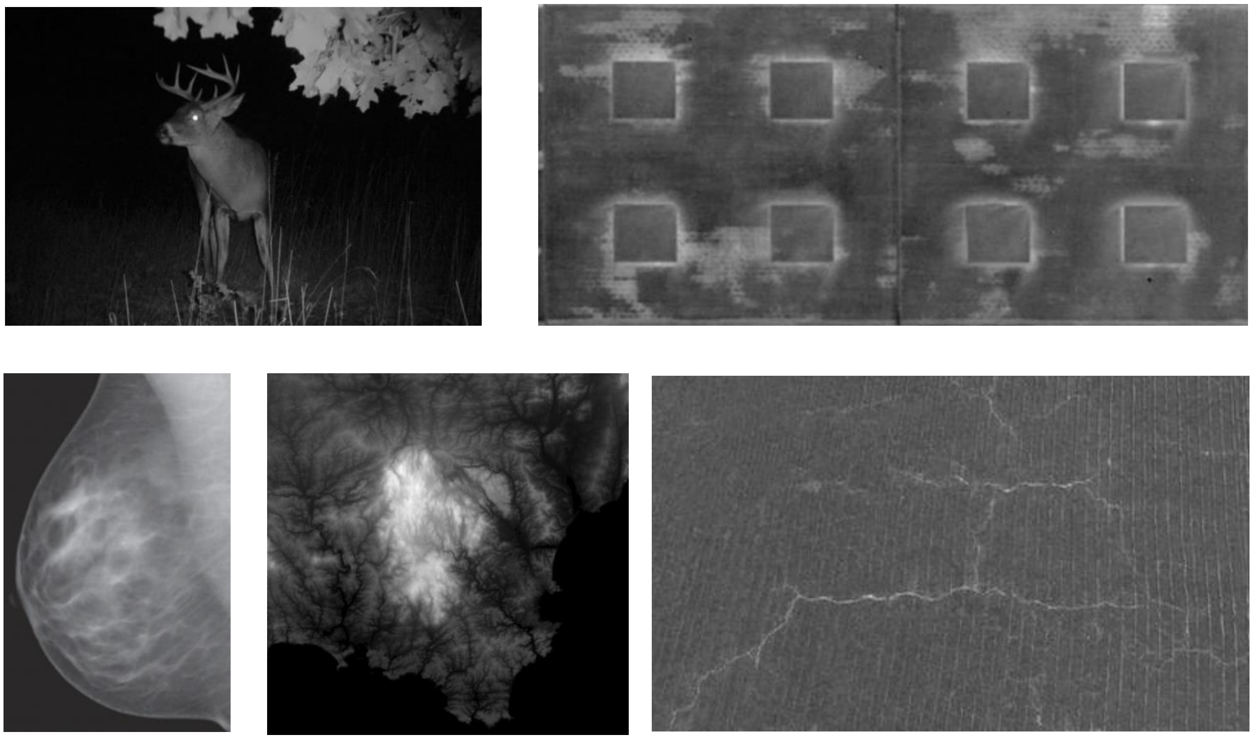

Figure 5. Apply your program to five images

<u>Submission</u>

To prepare for the submission of your work on Gradescope, create:

(1) a folder called “ps2-1,” that contains the following files:

- source code file(s)
- pseudocolor images created by your program:
    - night-vision-color.png
    - thermal-color.png
    - x-ray-color.png
    - topography-color.png

    - cracks-color.png
- "readme.txt" file that includes:
    - Operating system
    - IDE you used to write and run your code
    - The number of hours you spent to finish this problem

(2) a PDF file that contains the printouts and screenshots of all the files in the ps2-1 folder. (Include, if any, the mathematical derivation and/or description of your method in the PDF file. Handwritten notes should be scanned and included in the PDF file.)

Submit your work on Gradescope

Submit two files on Gradescope – replace “andrewid” with your own Andrew ID:

(1) **andrewid-ps2-files.zip** – this ZIP file should contain the ps2-1 folder and all the files requested.

(2) **andrewid-ps2-report.pdf** – this PDF file serves as the report of your work, and it should contain the printouts and screenshots of all the files in the “ps2-1” folder. (Include, if any, the mathematical derivation and/or description of your method in the PDF file. Handwritten notes should be scanned and included in the PDF file.)

Please organize pages with section titles and captions to make the report easy to read.