Image Processing and Learning Methods

Assignment 2

Due: 11:30 pm, Friday 03/05/2021 (Cleveland Time)

**100 points in total**

**General Requirements:**

1. Please place semicolons to avoid unnecessary console output, such as:

I=imread(‘1.jpg’);

2. In your main function, place a message “----Done for Question \*----” followed by a pause command (i.e., wait for a key to be pressed before continuing) at the end of each solution, where \* is the question number (1, 2, 3, …), such as:

*disp('-----Done for Question 2-----');*

*pause;*

3. Please submit your zipped .m files via the Blackboard system. The zipped file only has .m files without any image files.

**Grading:**

Q1-Q4: 25 points for each

**Questions:**

**1.** Read in the image (*Food.jpg*) and save it in an array **food**. Implement a **Mapping** function to **linearly** rescale the grayscale input image into a new intensity range. The prototype of this function should be:

**function[scaledIm, transfunc] = Mapping(Im, range)**

where Im is the original grayscale image, range is a vector containing the new range of the scaled image, and scaledIm is the rescaled (transformed) image. Make sure that your function shows an appropriate error message if the range contains the invalid data (out of range). Note: Both input and output images of the **Mapping** function should be an array with data type uint8.

Call the **Mapping** function to scale the image **food** into an appropriate range [newMin newMax] so the original image is enhanced to a good quality. Please show the original and scaled images in Figure 1 with appropriate titles.

**2.** Implement an **AverageFiltering** function to perform a filtering operation, as explained in class, on the input image. The prototype of this function should be:

**function [filteredIm] = AverageFiltering (im, mask)**

where im is the original grayscale image and mask is the square filter with odd numbers of rows and columns. Make sure that your function shows appropriate error messages when the mask’s dimension is not an odd number, the mask is not a square. Note: both input and output images of the **AverageFiltering** function should be an array with data type uint8. You are not allowed to simply use the Matlab “filter2” or “conv2” or “fspecial” or “imfilter” or other built-in functions in your implementation.

Call **AverageFiltering** function to process the noisy image ***Circuit.jpg*** using a **standard** 3×3 averaging filter and a **standard** 5×5 averaging filter, respectively. Display the original image and two processed images in Figure 2 with appropriate titles.

**3.** Implement a **MedianFiltering** function to perform a filtering operation, as explained in class, on the input image. The prototype of this function should be:

**function filteredIm = MedianFiltering(im, mask)**

where im is the original grayscale image and mask is the square filter with odd numbers of rows and columns. Make sure that your function shows appropriate error messages when the mask’s dimension is not an odd number and the mask is not a square. Note: both input and output images of the **MedianFiltering** function should be an array with data type uint8. You are not allowed to simply use the Matlab “medfilt2” or other built-in functions in your implementation. However, you can use the built-in sorting function to sort the values covered by the mask.

Call this function to process the same noisy image ***Circuit.jpg*** using a **standard** 3×3 median filter and a **standard** 5×5 median filter, respectively. Display the original image and two processed images in Figure 3 with appropriate titles.

Hint: you can use matlab built-in function median() to compute the median of an array.

**4.** Use a3×3 Laplacian mask to filter the image ***Moon.jpg*** by calling an **appropriate Matlab built-in function**. Use the formula **Enhanced Image = Original Image - Filtered Image** to get the final enhanced image (Hint: This formula ensures that one kind of Laplacian masks can be used). Use **imshow** to display four images including the original image, the filtered image, the scaled filtered image, and the enhanced image, in Figure 4 with appropriate titles.