CS 663 : Digital Image Processing : Assignment 1

Instructor : Suyash P. Awate

Due Date :-

Note: The input data / image(s) for a question is / are present in the corresponding data/ subfolder.

5 points are reserved for submission in the described format.

For all questions, display images using a colormap that uses at least 200 colors / intensities. Also, display the colormap alongside each image. Don't round the resulting images to integers; use a floating-point data type for the pixel value.

1. (15 points) Image Resizing.

   (a) (3 points) Image Shrinking.
   Input image: `1/data/circles_concentric.png`.
   Assume the pixel dimensions to be equal along both axes, i.e., assume an aspect ratio of 1:1 for the axes.
   Shrink the image size by a factor of d along each dimension using image subsampling by sampling / selecting every d-th pixel along the rows and columns.
   • Write a function `myShrinkImageByFactorD.m` to implement this.
   • Display the original and subsampled images, with the correct aspect ratio, for $d = 2$ and $d = 3$ appropriately to clearly show the Moire effects. Display the pixel units along each axis and the colorbar.

   (b) (6 points) Image Enlargement using Bilinear Interpolation.
   Input image: `1/data/barbaraSmall.png`.
   Assume the pixel dimensions to be equal along both axes, i.e., assume an aspect ratio of 1:1 for the axes. Consider this image as the data. Consider the number of rows as M and the number of columns as N.
   Resize the image to have the number of rows $= 3M - 2$ and the number of columns $= 2N - 1$, such that the first and last rows, and the first and last columns, in the original and resized images represent the same data.
   Use bilinear interpolation for resizing.
   • Write a function `myBilinearInterpolation.m` to implement this.
   • Display the original and resized images, without changing the aspect ratio of objects present in the image. Display the pixel units along each axis and the colorbar.

   (c) (6 points) Image Enlargement using Nearest-Neighbor Interpolation.
   Redo the previous problem using nearest-neighbor interpolation.

• Write a function `myNearestNeighborInterpolation.m` to implement this.

• Display the original and resized images.

2. (55 points)

   Input images:
   (1) `2/data/barbara.png`,
   (2) `2/data/TEM.png`,
   (3) `2/data/canyon.png`,
   (4) `2/data/retina.png`,
   (5) `2/data/church.png`,
   (6) `2/data/chestXray.png`, and
   (7) `2/data/statue.png`

   Image (4) has an associated binary image `2/data/retinaMask.png` indicating the foreground region. All the processing on image (4) should be performed only using the intensities in the foreground region. Image (4) also has an associated reference image `2/data/retinaRef.png` and its associated binary image `2/data/retinaRefMask.png` which are required for part (d) of the question only.

   Assume the pixel dimensions to be equal along both axes, i.e., assume an aspect ratio of 1:1 for the axes.

   For the color images, apply the analysis independently to each channel (Note: this is a suboptimal way of processing color images, in general; some of the reasons for which will be evident from the results that you will get).

   (a) **(2 points) Foreground Mask**

   Find a binary mask for the foreground region for image (7).

   • Write a function `myForegroundMask.m` to implement this.

   • Display the original image, the binary mask and the masked image.

   (b) **(3 points) Linear Contrast Stretching**

   Design a linear grayscale transformation function to enhance the intensity contrast such that the resulting intensity range is $[0, 255]$.

   • Write a function `myLinearContrastStretching.m` to implement this.

   • Show the formula (or the pseudo code) for the linear function in the report.

   • Display the original image and the contrast-enhanced image, without changing the aspect ratio of objects present in the image. Display the colorbar.

   • Show the above results on input images 1, 2, 3, 5, 6 and foreground region of image 7 (using the mask generated in part (a)).

   • Explain your observations after applying contrast stretching on image (5). Why do you think contrast stretching is or isn't effective here?

   (c) **(5 points) Histogram Equalization (HE)**

   Perform (global) HE on the entire image.

   • Write a function `myHE.m` to implement this.

   • Display the original image and the contrast-enhanced image.

- Show the above results on input images 1, 2, 3, 5, 6 and foreground region of image 7 (using the mask generated in part (a)).

- Explain your observations after applying HE on image (5). Which one would you prefer to improve image (5)- HE or contrast stretching?

(d) **(15 points) Histogram Matching (HM)** You are provided two images: input image `2/data/retina.png` and reference image `2/data/retinaRef.png`. Perform HM on the input image by matching the histogram with that of the reference image. Note that you don't need to show results for other images for this part of the question.

- Write a function `myHM.m` to implement this.
- Display the original image, histogram-matched image and the histogram-equalised image.
- State your observations.

(e) **(30 points) Contrast-Limited Adaptive Histogram Equalization (CLAHE)**
Perform CLAHE on the entire image. Manually tune the window-size parameter and the histogram-threshold parameter $\in [0,1]$ to an appropriate values in order to perform contrast enhancement for objects in the image, while preventing excessive noise amplification. For pixels close to the boundary, where the window falls outside the image, use only the pixels that are within the window and within the image (i.e, effectively cropping the window to restrict it within the image boundaries).

- Write a function `myCLAHE.m` to implement this.
- Display the original image and the contrast-enhanced image.

Redo CLAHE with neighborhood sizes chosen to be (i) significantly larger and (ii) significantly smaller than that chosen for the previous result in order to clearly show the effects of (i) low contrast improvement and (ii) excessive noise amplification.

- Display the additional two contrast-enhanced images.

Redo CLAHE with (i) the same window size as before and (ii) histogram-threshold parameter set to a value that is half of the value tuned before.

- Display the additional contrast-enhanced images.
- Show the above results on input images 1, 2, 3, 6.

3. (25 points)

Consider a (non-discrete) image $I(x)$ with a continuous domain and real-valued intensities within $[0,1]$. Let the image histogram be $h(I)$, with mass 1. Consider the histogram $h(I)$ is split into two histograms (i) $h_1(I)$ over the domain $[0,a]$ and (ii) $h_2(I)$ over the domain $(a,1]$, for some arbitrary $a \in (0,1)$. Assume that the histogram mass within $[0,a]$ is $\alpha \in (0,1)$.

(a) (8 points) Suppose you perform histogram equalization over the two intensity invervals $[0,a]$ and $(a,1]$ separately, in a way that preserved the masses of the two histograms $h_1(I)$ and $h_2(I)$ after the transformation. Derive the mean intensity for the resulting histogram (or, equivalently, image) and include it in the report.

(b) (2 points) Let the chosen intensity a be the median intensity for the original histogram $h(I)$. Assume that the mean intensity for the original histogram $h(I)$ is also a. Then, what is the mean intensity for the resulting histogram (or, equivalently, image). Show the derivations clearly in the report.

(c) (5 points) Describe a scenerio where the above described histogram-based intensity transform with *a = median intensity* will do a better job in intensity transformation than a simple histogram equalization. Explain the reasons clearly.

(d) (10 Points) Do an online search to find an image along the lines of your reasoning. Write a code for this intensity transformation and demonstrate the better performance on the image you obtained. Note that the better performance should be distinctly evident.