Digital Image Processing Homework 1

Jiajie Li 1750655

November 1st, 2020

Abstract

Li Hua is a pulmonologist at Tongji Hospital and has been testing for COVID-19. Since CT is one of the most commonly used tests for COVID-19, Li Hua and his department have to examine a large number of CT images every day. This is a report on the use of digital image processing technology to help Li Hua.

1 Introduction and Overview

Since CT is one of the most commonly used tests for COVID-19, Lihua and his department examine a large number of CT images every day. We need to use histogram equalization in the first question to make the contrast of CT images more appropriate, and interpolation in the second question to make the CT images scaled equally and become larger and clearer.

2 Problem 1: CT Contrast Enhancement

2.1 Problem 1-a

Calculate and plot the histogram of original images and briefly comment on the shape of the histograms.

2.1.1 Computational Results

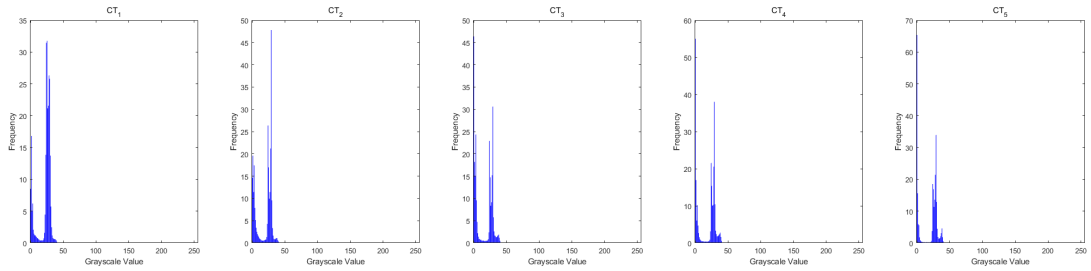

Figure 1: Computational Results of Problem 1-a

2.1.2 Theoretical Background

Histograms are widely used in many computer vision applications to detect changes in a scene in a video by marking significant statistical changes in edge and color from frame to frame. Each point of interest is "labeled" by a histogram of similar features, which identifies the point of interest in the image. Histograms of edges, colors, angles, etc., form a generic feature type that can be passed to a target recognition classifier. Histogram sequences of colors and edges can also be used to identify whether a web video has been copied.

Suppose there is a matrix containing information about an image (grayscale values 0 - 255), and since

the range of numbers is known to contain 256 values, the range can be divided into subregions (i.e., bins) according to certain rules. For example:

$$\begin{aligned} [0, 255] &= [0, 15] \cup [16, 31] \cup \cdots \cup [240, 255] \\ \text{range} &= \text{bin}_1 \cup \text{bin}_2 \cup \cdots \cup \text{bin}_{n=15} \end{aligned} \tag{1}$$

2.1.3 Comments

The distribution is very uneven, the right side is almost non-existent, the high frequencies are concentrated on the left side, and the apparent performance is poor photo contrast.

2.2 Problem 1-b

Apply global histogram equalization to original images. Submit the modified images. Check the histogram of the modified images grayscale values and comment on visually desirable/undesirable regions in the modified image.

2.2.1 Computational Results

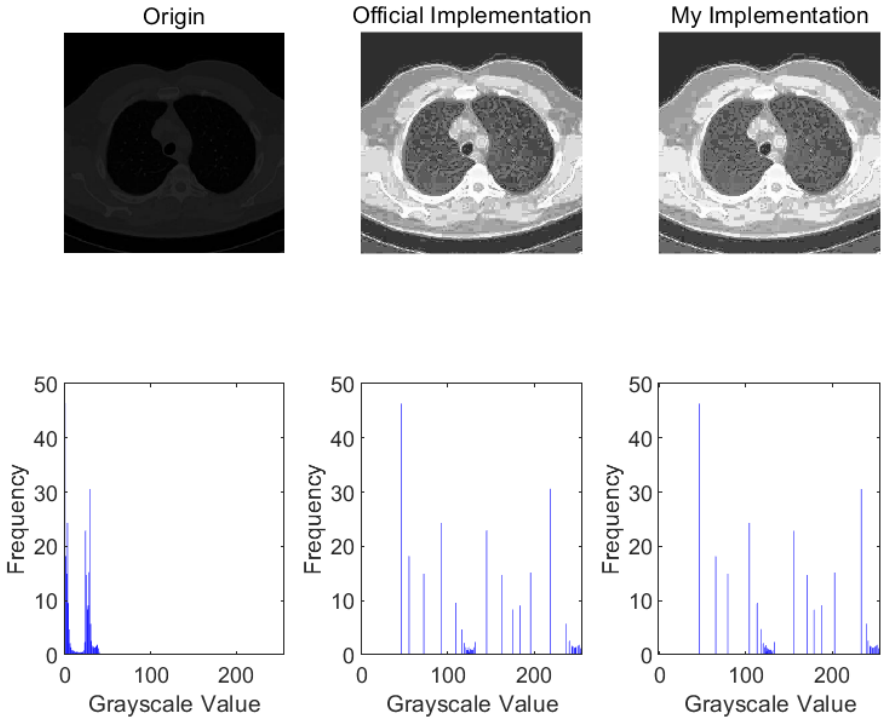

Figure 2: Computational Results of Problem 1-b

2.2.2 Theoretical Background

Histogram equalization is often used to increase the global contrast of many images, especially when the images are quite close in contrast to the useful data. By this method, the brightness is better distributed across the histogram. This can then be used to enhance the local contrast without affecting the overall contrast, and histogram equalization does this by effectively extending the commonly used brightnesses.

Set the variable r to represent the gray level of pixels in the image. If we normalize the gray level, then

$0 \leq r \leq 1$, where $r = 0$ means black and $r = 1$ means white. For a given image, the gray level of each pixel value in $[0, 1]$ is random. The grayscale distribution of an image is represented by the probability density function $P_r(r)$.

The functional expression for the image for histogram equalization is:

$$S_i = T(r_i) = \sum_{i=0}^{k-1} \frac{n_i}{n} \tag{2}$$

2.2.3 Comments

The histogram equalization method is useful if an image is dark or light overall. However, histogram equalization is a global process, which is indiscriminate in the data it processes and may increase the contrast of background noise and decrease the contrast of useful signals. In addition, the gray level of the image decreases after equalization, and some detail will be lost; the peaks of the histogram will be unnaturally and excessively contrasted after equalization; some areas of the CT image have good contrast and others do not, so global histogram equalization may not be the best. There is some whitening in the background that we don't care about, but there are some black masses in the lungs that we do care about.

2.3 Problem 1-c

Apply locally adaptive histogram equalization to original images. Submit the modified images. Check the histogram of the modified images grayscale values. Choose and report the number of tiles and the clipping limit for attaining higher contrast while avoiding the generation of noisy regions and the amplification of nonuniform lighting effects. Describe the subjective quality of the modified image compared to the result in (b)

2.3.1 Theoretical Background

Flaws in HE Histogram equalization (HE) is a very commonly used histogram class method, the basic idea is to determine a mapping curve from the gray-scale distribution histogram of the image, which is used to perform a gray-scale transformation of the image in order to improve the contrast of the image. The mapping curve is actually the cumulative distribution histogram (CDF) of the image. However, HE is a global adjustment method, which is not effective in improving local contrast and may be very poor in some cases.

AHE The reason for this is that using HE on the original image causes the histogram to be shifted to the right as a whole and does not take full advantage of the entire grayscale dynamic range. In order to improve the local contrast of the image, it has been proposed to divide the image into sub-blocks and apply HE to the sub-blocks, which is AHE (Adaptive Histogram Equalization).

Move template W line by line on image A and the center $c(x_0, y_0)$ of template W corresponds to the point $f(x_0, y_0)$ on the image; calculate the histogram equilibrium change relationship for the region of template $W : g(x, y) = T(f(x, y)$, calculate the equilibrium corresponding pixel value for the center $c((x_0, y_0))$ of template $W : g((x_0, y_0)) = T(f(x_0, y_0)))$. . Replace $f((x_0, y_0))$ with $g((x_0, y_0))$; a line-by-line calculation is performed to obtain an adaptive histogram equalization image of the whole image.

CLAHE From the result image, we can see that the local contrast has been improved, and the visual effect is better than the HE, but there is still a problem: AHE improves the local contrast too much, which leads to image distortion.

Therefore, we need to crop the histogram in the bin so that its amplitude is below a certain upper limit (ClipLimit), but we can't throw away the crop, we also need to distribute the crop evenly over the entire grayscale interval to ensure that the total area of the histogram remains unchanged, as shown below.

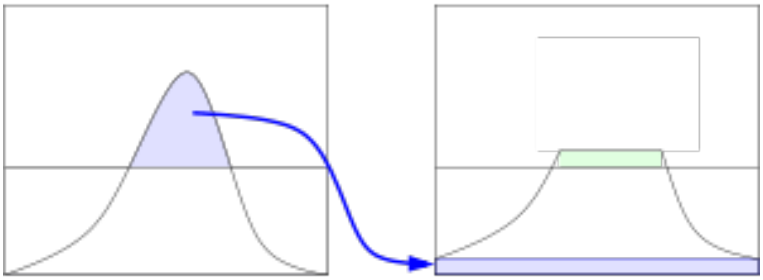

Figure 3: ClipLimit Diagram

As you can see, the histogram will now rise another height overall, seemingly beyond the upper limit we set. There are many ways to achieve this, you can repeat the cropping process several times to make the rise insignificant, or use another common method.

Set the clipping value to $ClipLimit$, find the sum of the parts of the histogram above that value and $totalExcess$, assuming that $totalExcess$ is divided equally among all gray levels, find the height of the overall rise of the histogram resulting from this $L = totalExcess/N$, and process the histogram using $upper = ClipLimit - L$ as the bound as follows.

1. If the amplitude is higher than ClipLimit, set it directly to ClipLimit.
2. If the magnitude is between Upper and ClipLimit, fill it to ClipLimit.
3. If the amplitude is lower than the Upper, fill L pixels directly.

After the above operation, the number of pixels used to fill will usually be slightly less than the totalExcess, which means that there are still some remaining pixels left to be divided, and this remaining comes from (1) and (2). We can then divide these evenly between those pixels whose current magnitude is still less than ClipLimit's grayscale value.

2.3.2 Pseudocode

Algorithm 1: Local Histogram Equilibrium

```
create a new image with all pixel values to be zero
while number of iterations not reached do
    select the next area by step size
    histogram equalize(With CLAHE) the region
    add up the values of the pixels to the new image
end while
Divide the value of each pixel of the new image by the number of times it has been counted.
return
```

2.3.3 Degree of Overlap

The partial histogram equalization algorithm, also known as the sub-block histogram equalization algorithm is classified according to the degree of overlap of the equalized sub-blocks, which can be divided into three kinds of sub-block non-overlap, sub-block overlap and sub-block partial overlap, I conducted several experiments and chose the sub-block partial overlap equalization algorithm, the difference between this method (POSHE) and the sub-block overlap method is described as below:

1. Subblocks are not moved pixel by pixel, but are moved in steps that are approximately a fraction of the size of the subblock.

2. The grayscale conversion function for sub-block equalization is used to map not only the grayscale values of the center pixel of the sub-block, but also the grayscale values of all the pixels in the sub-block.

3. For a pixel that has been equalized multiple times, the equalization result is averaged as the grayscale value of the pixel in the output image.

The sub-block partial overlap algorithm is characterized by the following:

1. As the partial overlapping of sub-blocks reduces the difference in the shape of the equilibrium function between adjacent sub-blocks, the block effect is basically eliminated.

2. As the total number of sub-block equilibrium times is much less than the sub-block overlap method, the computational efficiency is greatly improved.

3. The image detail enhancement ability is similar to that of the sub-block overlap algorithm.

2.3.4 Computational Results

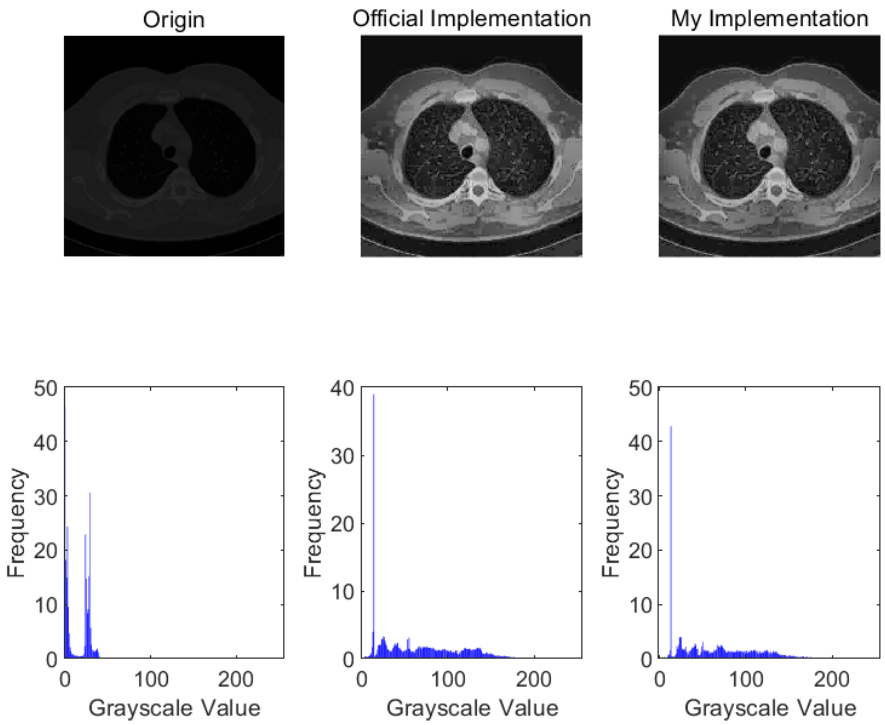

Figure 4: Computational Results of Problem 1-c

2.3.5 Comments

Unlike the normal histogram equalization algorithm, the AHE algorithm varies the image contrast by calculating the local histogram of the image and then redistributing the brightness. As a result, the algorithm is more suitable for improving the local contrast of the image and obtaining more image detail.

To get a higher contrast while avoiding creating noisy areas and amplifying nonuniform lighting effects. I chose tiles of $[5, 5]$, *ClimbLimit* of 0.05 and a step scale of 0.05. Above are the results of my experiments, which I subjectively believe yielded more local detail.

2.4 Problem 1-d

Apply a γ-nonlinearity mapping to each image to perform contrast enhancement. Submit the modified images. For each image, find and report a value of γ that allows you to see more details.

2.4.1 Theoretical Background

RGB value and power is not a simple linear relationship, but a power function relationship, the exponent of this function is called the Gamma value, usually 2.2, and this conversion process, called the Gamma correction.

$$V_{\text{out}} = AV_{\text{in}}^{\gamma} \tag{3}$$

Where A is a constant and the input and output values are non-negative real values. Generally in the usual case where A = 1, the range of input and output values is between 0 and 1.

2.4.2 Lists of γ

	Name	γ
1	CT-1	0.9
2	CT-2	0.4
3	CT-3	0.5
4	CT-4	0.8
5	CT-5	1.2

Table 1: The γ value for each image.

2.4.3 Computational Results

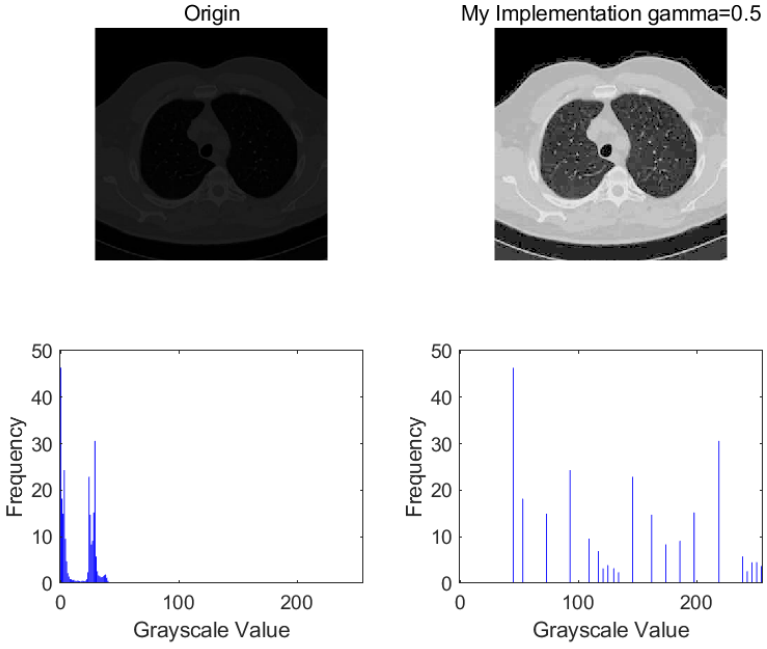

Figure 5: Computational Results of Problem 1-d

3 Problem 2: Resizing X-ray Images

3.1 Problem 2-a

Apply the nearest neighboring interpolation methods to X-ray images. Submit the modified images. Check the images and comment on the desirable/undesirable regions in the modified image.

3.1.1 Theoretical Background

Nearest neighboring interpolation is the simplest kind of interpolation algorithm, when the image is enlarged, the missing pixels are generated by directly using the nearest original color, i.e. mirroring the pixels next to them. In the four neighboring pixels of the pixel to be sought, the gray level of the closest neighbor to the pixel to be sought is assigned to the pixel to be sought.

3.1.2 Computational Results

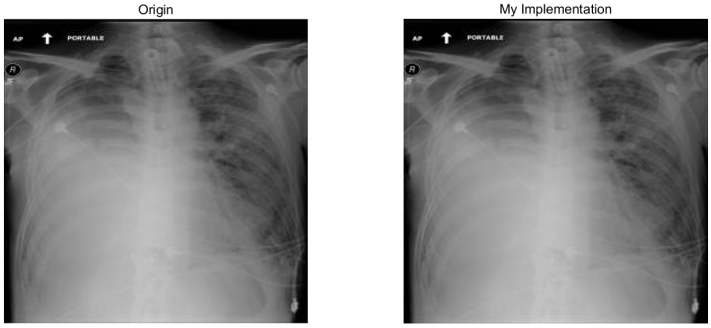

Figure 6: Computational Results of Problem 2-a

3.1.3 Comments

This is the most basic and simple image scaling algorithm, the effect is also the worst, the magnified image has a very serious mosaic, the reduced image has a very serious distortion; the root cause of the effect is the simple method of the closest interpolation introduced serious image distortion, for example, when the coordinates of the source map obtained by the target map inverse of the coordinates is a floating point, the use of rounding methods, directly using the closest and the value of the floating point pixel, this method is very unscientific!

3.2 Problem 2-b

Apply the bi-linear neighboring interpolation methods to X-ray images. Submit the modified images. Comment on the subjective quality of the modified image compared to the result in (a).

3.2.1 Theoretical Background

As shown in Fig.8, Q_{12}, Q_{22}, Q_{11}, Q_{21} are known, but the point to be interpolated is point P. This is called bilinear interpolation, first interpolate two points R_1 and R_2 in the x-axis direction, then interpolate point P according to R_1 and R_2, this is called bilinear interpolation.

We know the value of the red data point, and we obtain the value of the green data point by bilinear interpolation.

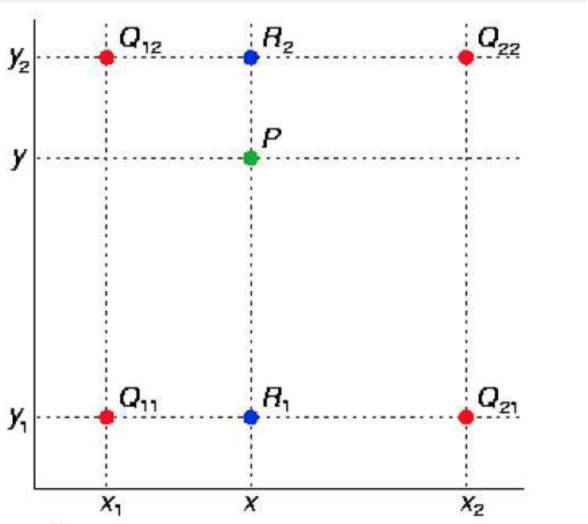

Figure 7: Computational Results of Problem 2-c

Suppose we want to get the value of the unknown function f at the point $P = (x, y)$, and suppose we know the value of the function f at the four points $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, and $Q_{22} = (x_2, y_2)$.

First, linear interpolation is performed in the x-direction to obtain:

$$\begin{array}{lll} f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) & \text{Where} & R_1 = (x, y_1) \\ f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) & \text{Where} & R_2 = (x, y_2)\,. \end{array} \tag{4}$$

Then linearly interpolate in the y-direction to obtain the:

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \tag{5}$$

This gives the desired result, $f(x, y)$,

$$\begin{array}{l} f(x,y) \approx \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y_2 - y) + \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y_2 - y) \\ + \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y - y_1) + \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y - y_1) \end{array} \tag{6}$$

If a coordinate system is chosen such that the coordinates of the four known points of f are (0, 0), (0, 1), (1, 0), and (1, 1), then the interpolation formula is reduced to the following:

$$f(x,y) \approx f(0,0)(1 - x)(1 - y) + f(1,0)x(1 - y) + f(0,1)(1 - x)y + f(1,1)xy \tag{7}$$

Or expressed as a matrix operation:

$$f(x,y) \approx \begin{bmatrix} 1 - x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1 - y \\ y \end{bmatrix} \tag{8}$$

3.2.2 Comments

The results of this method of interpolation are usually not linear, and the results of linear interpolation are independent of the order in which they are interpolated. Interpolation in the y-direction first, followed by interpolation in the x-direction, yields the same result.

The bilinear interpolation method is more complex than the nearest point method, is more computationally intensive but does not have the disadvantage of grayscale discontinuity, and yields largely satisfactory results. It has a low-pass filtering property that impairs the high-frequency component and the image contours may be a little blurred.

3.2.3 Computational Results

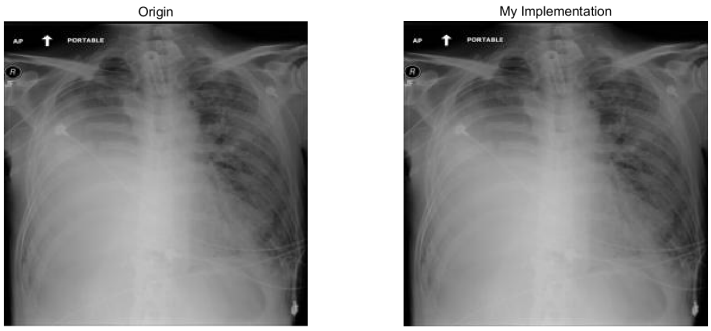

Figure 8: Computational Results of Problem 2-b

3.3 Problem 2-c

Apply the bi-cubic neighboring interpolation methods to X-ray images. Submit the modified images. Compare the modified images to the result problem (a) and (b).

3.3.1 Theoretical Background

Bi-quadratic interpolation is the most commonly used interpolation method in two-dimensional space. In this method, the value of the function f at point (x,y) can be obtained by a weighted average of the nearest sixteen samples in a rectangular grid, where two polynomial interpolation cubic functions are used, one in each direction. BiCubic base functions based on bi-quadratic interpolation of the BiCubic base function of the following form(where a takes -0.5):

$$W(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & \text{for } |x| \leq 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

For the pixel (x, y) to be interpolated (x and y can be floating-point numbers), take the nearby 4x4 neighborhood points (x_i, y_j), $i, j = 0, 1, 2, 3$. Interpolate according to the following formula:

$$f(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} f\left(x_i, y_j\right) W\left(x - x_i\right) W\left(y - y_j\right) \tag{10}$$

3.3.2 Comments

Bicubic interpolation usually produces the best and most accurate interpolation graphs compared to the first two, but it's also almost always the slowest.

3.3.3 Computational Results

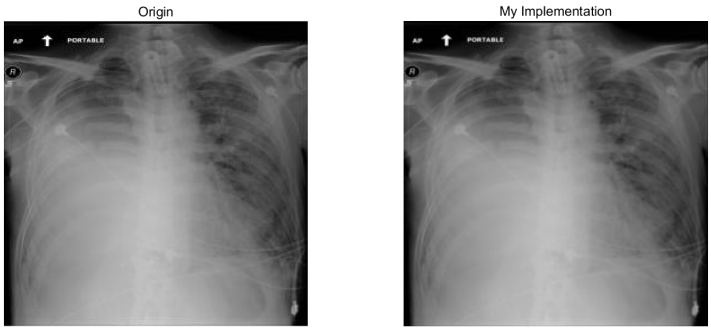

Figure 9: Computational Results of Problem 2-c

4 Appendix

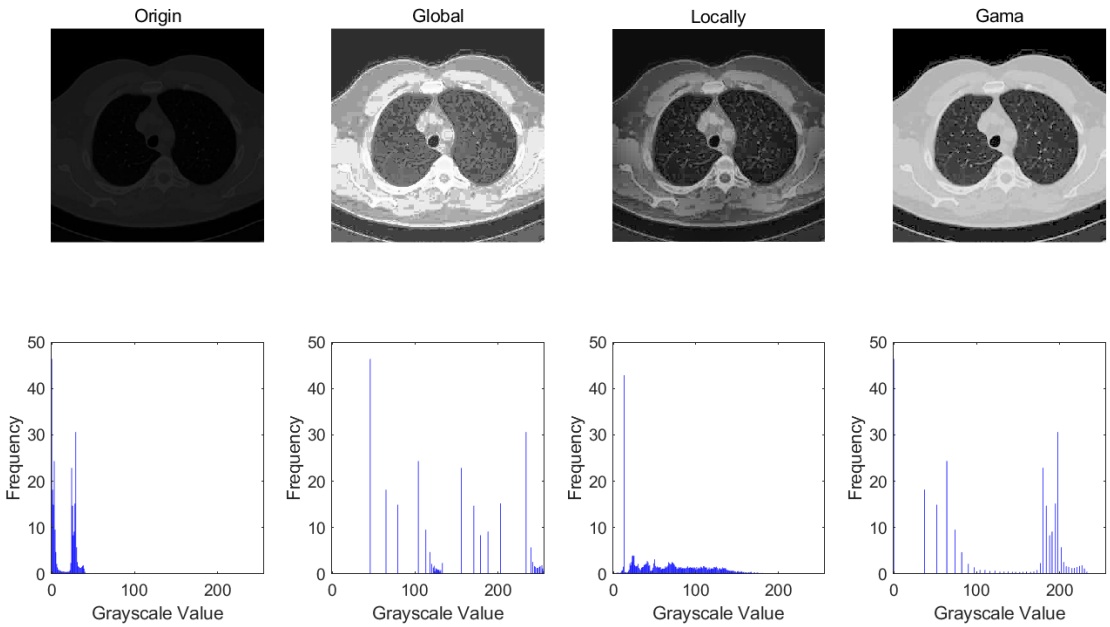

Figure 10: Comparison of the effects of the three methods of Task 1