Department of Computer Engineering

Bu-Ali Sina University

Digital image processing Course

Digital image processing technique for image enhancement , assignment 1

By:

Ehsan Rostami

Academic Supervisor:

Professor Hassan Khotanlou

Fall 2024

1 INTRODUCTION

Digital image processing (DIP) techniques are essential for enhancing visual information, which is critical in fields such as computer vision, medical imaging, and satellite imaging. This report presents an analysis of various DIP methods, including image zooming, contrast enhancement, histogram equalization, and spatial transformations applied to sample images. Each technique's effect on image quality, detail retention, and computational efficiency was evaluated through experiments conducted using Python and OpenCV. The results, accompanied by visual examples and a detailed analysis of each method's strengths and limitations, provide insight into their practical applications and suitability for different scenarios.

METHODOLOGY

In each exercise, the images were processed using specific techniques. The following methods were employed:

Image Zooming (Question 1)

- **Pixel Replication:** Enlarges the image by duplicating pixel values in blocks. This straightforward technique is computationally efficient but often produces a blocky, pixelated appearance.
- **Zero-Order Hold:** Similar to pixel replication but considers neighboring pixels for smoother transitions, reducing blockiness.
- **Implementation:** The two techniques were applied to `einstein.jpg` with a zoom factor of 2. The Python code developed for each method included comments explaining the steps of pixel manipulation.

Contrast Enhancement (Question 2)

- **Logarithmic Correction:** Enhances dark areas of an image by applying a logarithmic transformation, which brightens lower intensities. A scaling constant, c, was chosen to keep pixel values within displayable limits.
- **Gamma Correction:** Adjusts brightness based on an exponential transformation where γ controls intensity; values below 1.0 brighten, while values above 1.0 darken.
- **Implementation:** The methods were applied to `low_contrast.jpg` with histograms plotted to compare intensity distributions before and after enhancement.

Histogram Equalization (Question 3)

Redistributes pixel intensities to balance contrast and spread the values evenly across the intensity range. This technique is widely used to improve visibility in poorly contrasted images.

- **Implementation:** Histogram equalization was performed on `lena_gray.gif`, followed by plotting histograms of the original and equalized images.

Spatial Transformations (Question 4)

- **Translation:** Shifts the image by moving pixels along the x and y axes.
- **Rotation:** Rotates the image by a specified degree, here 60 degrees counterclockwise.
- **Implementation:** OpenCV's affine transformation functions were used to apply translation and rotation to `square.gif`.

Image Modification (Question 5)

Various mathematical transformations were applied to `skeleton.gif`, such as thresholding and edge enhancement, to observe the effect on clarity and contrast.

EXPERIMENTAL RESULTS

This section presents the detailed experimental results of each exercise question, with each sub-question answered separately. Figures are included to illustrate each stage of processing, and the results are analyzed based on advantages and disadvantages.

Question 1: Image Zooming

A) Describe Pixel Replication and Zero-Order Hold Methods for Image Zooming

- **Pixel Replication:** A method that duplicates pixels in blocks based on the zoom factor. It's straightforward and efficient but often results in a pixelated appearance due to blocky edges.
- **Zero-Order Hold:** An interpolation approach that considers neighboring pixels to create smoother transitions. It's a slight improvement over direct replication, reducing artifacts but can still introduce some blur.

B) Implementation of Zooming in 'einstein.jpg' Using Python

- **Figure 1a:** Original image `einstein.jpg`
- **Figure 1b:** Zoomed image with Zero-Order Hold (zoom factor 2)
- **Figure 1c:** Zoomed image with Pixel Replication (zoom factor 2)

Figure 1: 1a,1b,1c

The code for each method replicated pixels based on the zoom factor, either by direct duplication for Pixel Replication or by interpolating with neighboring pixels in Zero-Order Hold. OpenCV was used to manage image I/O, and custom functions were implemented to illustrate the zoom process.

C) Advantages and Disadvantages of Each Method

- **Pixel Replication:**
  - **Advantage:** Simple and computationally efficient.
  - **Disadvantage:** Leads to pixelated, blocky images, especially at higher zoom factors.
- **Zero-Order Hold:**
  - **Advantage:** Produces smoother edges compared to Pixel Replication.
  - **Disadvantage:** Introduces some blurriness and still lacks the detail restoration needed for high-resolution applications.

D) Display of Zoomed Images and Analysis of Increased Zoom Factor

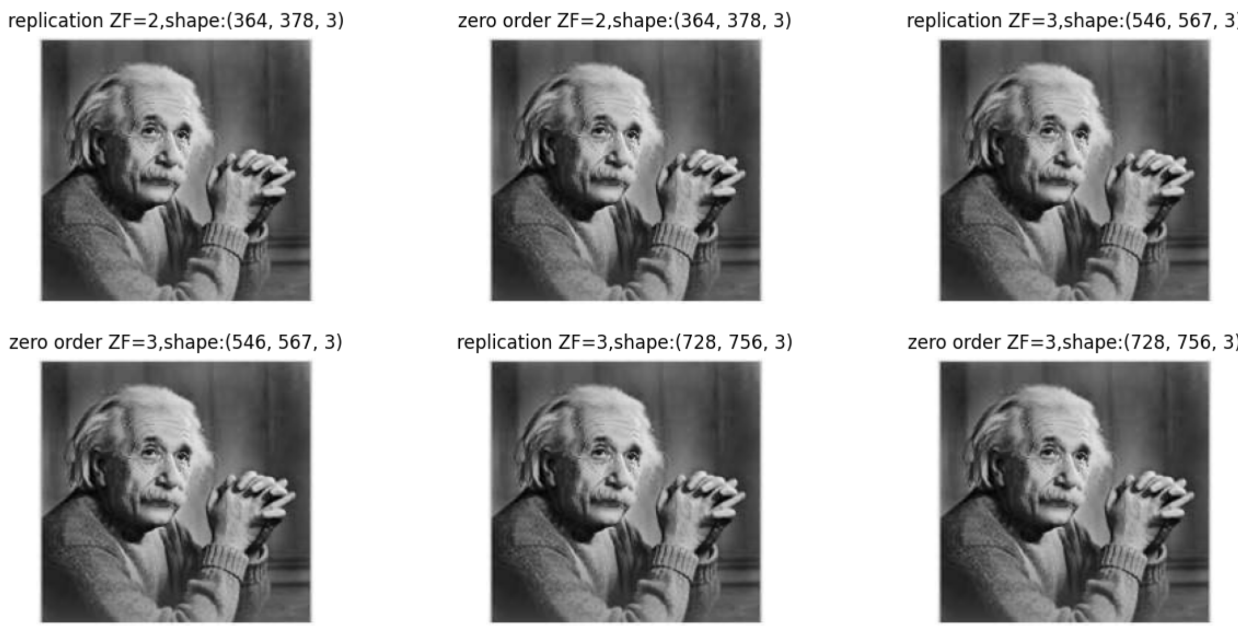

Figure 2: zoom factors impacts

With a zoom factor increased to 4, both methods amplified artifacts. Pixel Replication displayed larger blocks, while Zero-Order Hold introduced significant blurring. This highlights the limitations of these methods for large-scale zooming, where advanced interpolations like bilinear or bicubic methods are preferable.

E) Suggested Improved Method

An improved method could be bicubic interpolation, which considers more surrounding pixels for each interpolated pixel, yielding a smoother and more detailed result. Compared to Pixel Replication and Zero-Order Hold, bicubic interpolation can better preserve details, particularly at higher zoom levels, making it ideal for applications requiring image clarity.

Question 2: Contrast Enhancement

A) Normalize the Image

The image `low_contrast.jpg` was normalized by scaling pixel values to the 0 to 1 range, ensuring that both the Logarithmic and Gamma Correction methods work effectively.

B) Describe Logarithmic and Gamma Correction Methods for Contrast Stretching

- **Logarithmic Correction:** Enhances contrast by applying a logarithmic transformation, which brightens darker regions and compresses the intensity range of lighter areas.
- **Gamma Correction:** Adjusts image brightness by applying an exponential transformation, where γ controls intensity. This method allows targeted contrast control in mid-tones.

C) Application of Logarithmic Correction

The following formula was used:

$$\text{enhanced\_pixel\_value} = c \cdot \log(1 + \text{original\_pixel\_value})$$

where c was chosen to scale values within 0 to 1. The logarithmic enhancement improved visibility in the image's darker regions, though brighter areas remained relatively unchanged.

- **Figure 3a:** Logarithmic image `low_contrast.jpg`
- **Figure 3b:** Gamma image

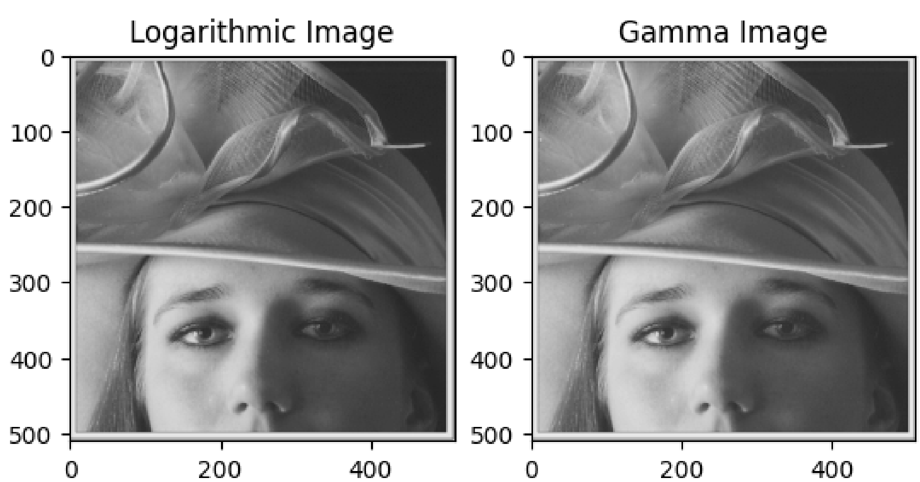

Figure 3: 3a,3b

D) Application of Gamma Correction

Gamma correction was applied with γ set to 0.5 to brighten the image's mid-tones. This provided a balanced enhancement across the intensity range.

- **Figure 4:** Gamma correction result with histogram

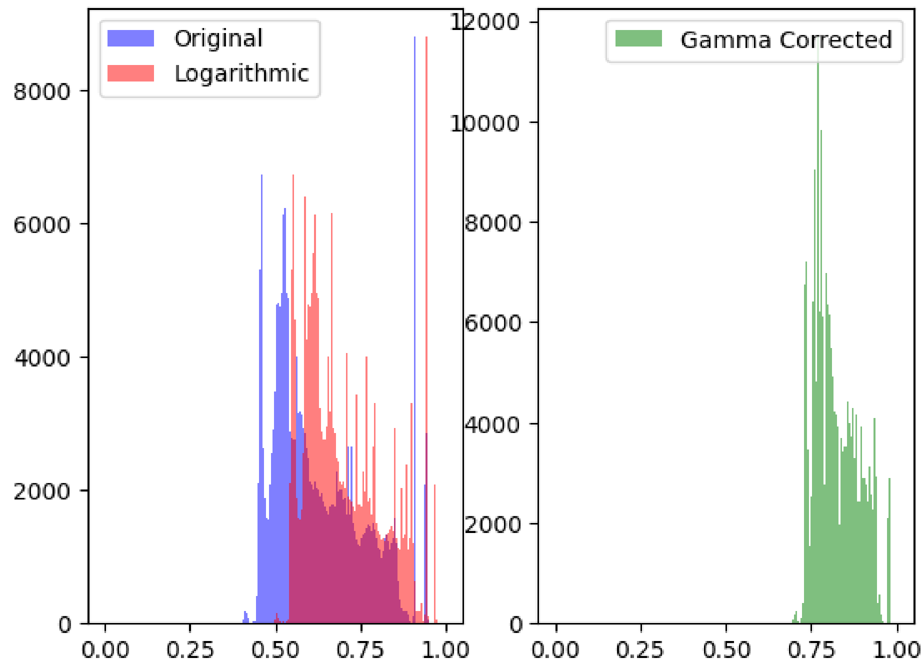

Figure 4: Gamma correction result histogram

E) Histogram Comparison and Analysis

- **Logarithmic Correction:**
  - **Advantage:** Enhances dark regions, useful for images with lost details in shadows.
  - **Disadvantage:** Limited effect on high-intensity areas, leaving highlights compressed.
- **Gamma Correction:**
  - **Advantage:** Offers flexibility in controlling brightness and contrast through the gamma parameter, improving mid-tone visibility.
  - **Disadvantage:** Requires careful gamma selection; may not be as effective for images needing significant contrast in high or low intensities.

Figure 4 shows the histograms, indicating better distribution of intensities with both corrections, though gamma correction achieved a more uniform enhancement.

Question 3: Histogram Equalization

A) Image Normalization

The `lena_gray.gif` image was normalized to ensure pixel values were within the 0 to 1 range before equalization.

B) Histogram Calculation and Analysis

- **Figure 5:** Original histogram of `lena_gray.gif`

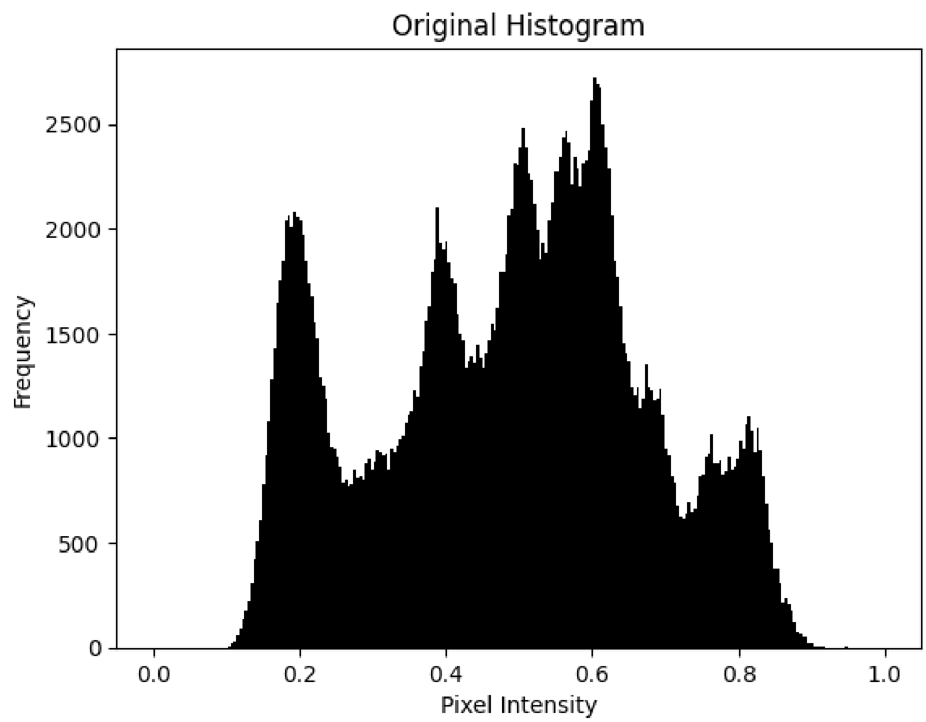

Figure 5: original histogram

The original histogram indicated uneven intensity distribution, with certain ranges dominating. This imbalance prompted the need for histogram equalization to stretch intensity values across the full range.

C) Histogram Equalization Algorithm

The algorithm calculated the cumulative distribution function (CDF) to map pixel intensities, re-distributing values for a more uniform contrast.

D) Display of Equalized Image and Histogram

- **Figure 6:** Histogram-equalized image and histogram

The equalized histogram showed a balanced spread across intensity levels, with both light and dark areas more visible.

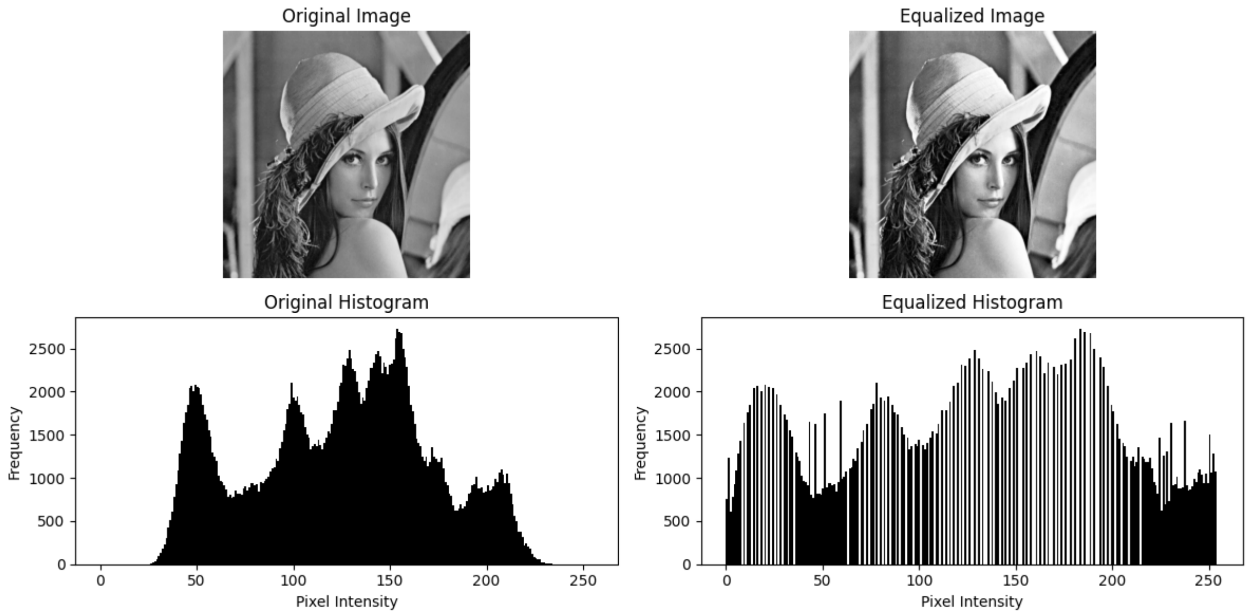

Figure 6: Histogram-equalized image and histogram

E) Analysis and Comparison of Original and Equalized Image

- **Histogram Equalization:**
  - **Advantage:** Effectively balances contrast, particularly beneficial for low-contrast images.
  - **Disadvantage:** May amplify noise in uniform areas, causing artifacts.

Question 4: Spatial Transformations

A) Translation of Small Square

The small square in `square.gif` was shifted 40 pixels along both the x and y axes, maintaining its intensity and structure.

B) Rotation of Large Square

The large square was rotated 60 degrees counterclockwise. A rotation matrix centered on the square was used, and the `warpAffine` function in OpenCV performed the transformation.

- **Figure 7a:** Original image `square.gif`
- **Figure 7b:** Translated and rotated image

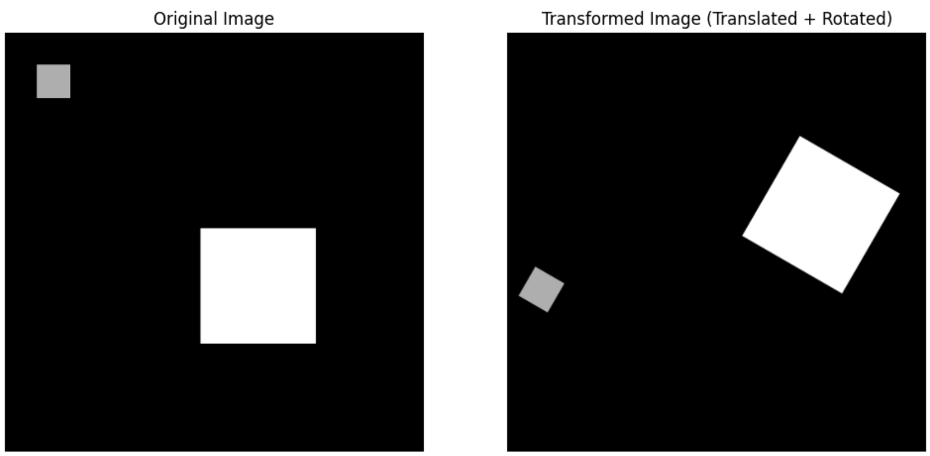

Figure 7: 7a,7b

Analysis of Transformation Results

- **Translation:**
  - **Advantage:** Simple manipulation ideal for shifting objects without altering their structure.
  - **Disadvantage:** Limited to 2D shifts; perspective distortion isn't considered.
- **Rotation:**
  - **Advantage:** Aligns objects effectively, useful in image registration.
  - **Disadvantage:** May introduce aliasing effects, visible as jagged edges at rotated boundaries.

Question 5: Image Modification on 'skeleton.gif'

A) Application of Mathematical Functions

Several transformations, including edge enhancement and thresholding, were applied to assess their impact on clarity and structure visibility.

Transformation 1: Exponential Transformation

Formula: $g(x,y) = e^{f(x,y)}$

The exponential transformation amplifies high-intensity values, making bright areas in the image more prominent. This transformation is useful for emphasizing brighter regions, although very bright areas may become saturated, potentially obscuring finer details.

Observation: The result of the exponential transformation highlights bright structures in the skeleton image, creating a stark contrast between bright and dark regions.

Transformation 2: Logarithmic Transformation

Formula: $g(x,y) = c \cdot \log_{10}(f(x,y) + 1)$

The logarithmic transformation compresses the dynamic range, enhancing details in darker regions more significantly than in brighter ones. This transformation is particularly useful for images with large dark areas where detail might otherwise be lost.

Observation: The logarithmic transformation reveals subtle details in the darker parts of the skeleton structure, offering a balanced contrast that improves visibility without making bright areas overly intense.

Transformation 3: Piecewise Transformation

Formula:

$$g(x,y) = \begin{cases} f(x,y) & \text{if } f(x,y) < 0.1 \\ 2 \cdot f(x,y) & \text{otherwise} \end{cases}$$

The piecewise transformation selectively enhances mid-range and high-intensity values, doubling pixel intensities above a threshold of 0.1 while leaving lower intensities unchanged. This provides a balanced contrast adjustment, enhancing brighter regions while preserving background details.

Observation: The piecewise transformation enhances mid-tone and bright areas, making the skeleton structure more visible without excessively amplifying the brightest or darkest parts of the image.

Comparison

Each transformation modifies the image uniquely, making them suitable for different applications:

- **Exponential Transformation**: Best for highlighting bright regions, though it may saturate high intensities.
- **Logarithmic Transformation**: Ideal for enhancing dark areas, suitable for images with low-intensity details.
- **Piecewise Transformation**: Selectively enhances mid-range intensities, providing a controlled contrast adjustment.

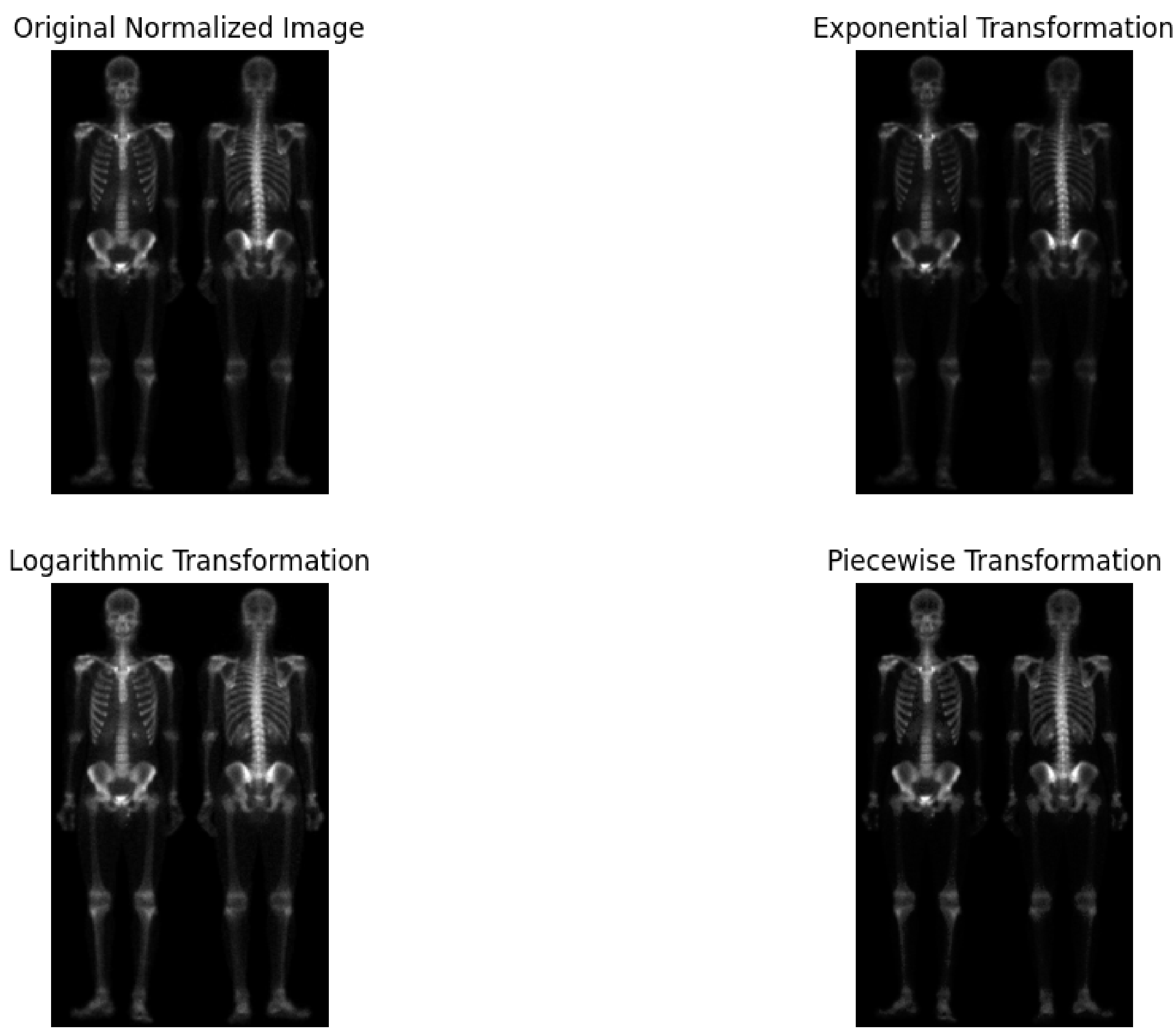

Figure 8: transformation results

CONCLUSION

Each image processing technique in this exercise has unique strengths and limitations, depending on the image characteristics and application requirements. The experiments highlighted:

- **Image Zooming:** Both Pixel Replication and Zero-Order Hold have limitations, especially at high zoom factors. Bicubic interpolation could provide smoother results for high-resolution applications.
- **Contrast Enhancement:** Logarithmic correction and gamma correction effectively improve low-contrast images, with gamma correction offering more control over brightness adjustments.
- **Histogram Equalization:** Redistributes intensities effectively, enhancing contrast in low-contrast images, though it may over-emphasize noise.
- **Spatial Transformations:** Translation and rotation enable object repositioning, but rotation may benefit from anti-aliasing to reduce jagged edges.
- **Image Modification:** Edge enhancement produced the clearest result for `skeleton.gif`, underscoring its suitability for tasks requiring sharp boundaries.

These findings illustrate the value of method selection in digital image processing and lay a foundation for future exploration of more advanced techniques like bicubic interpolation and adaptive histogram equalization for higher image quality and detail retention.