**CS5500 – Digital Image Processing**

**Cal Poly Pomona**

**Homework 1**

**Fall 2023**

**Description:**

Spatial Resolution

Gray Level Resolution

Name: Fidelis Prasetyo

Email: ([fprasetyo@cpp.edu](mailto:fprasetyo@cpp.edu))

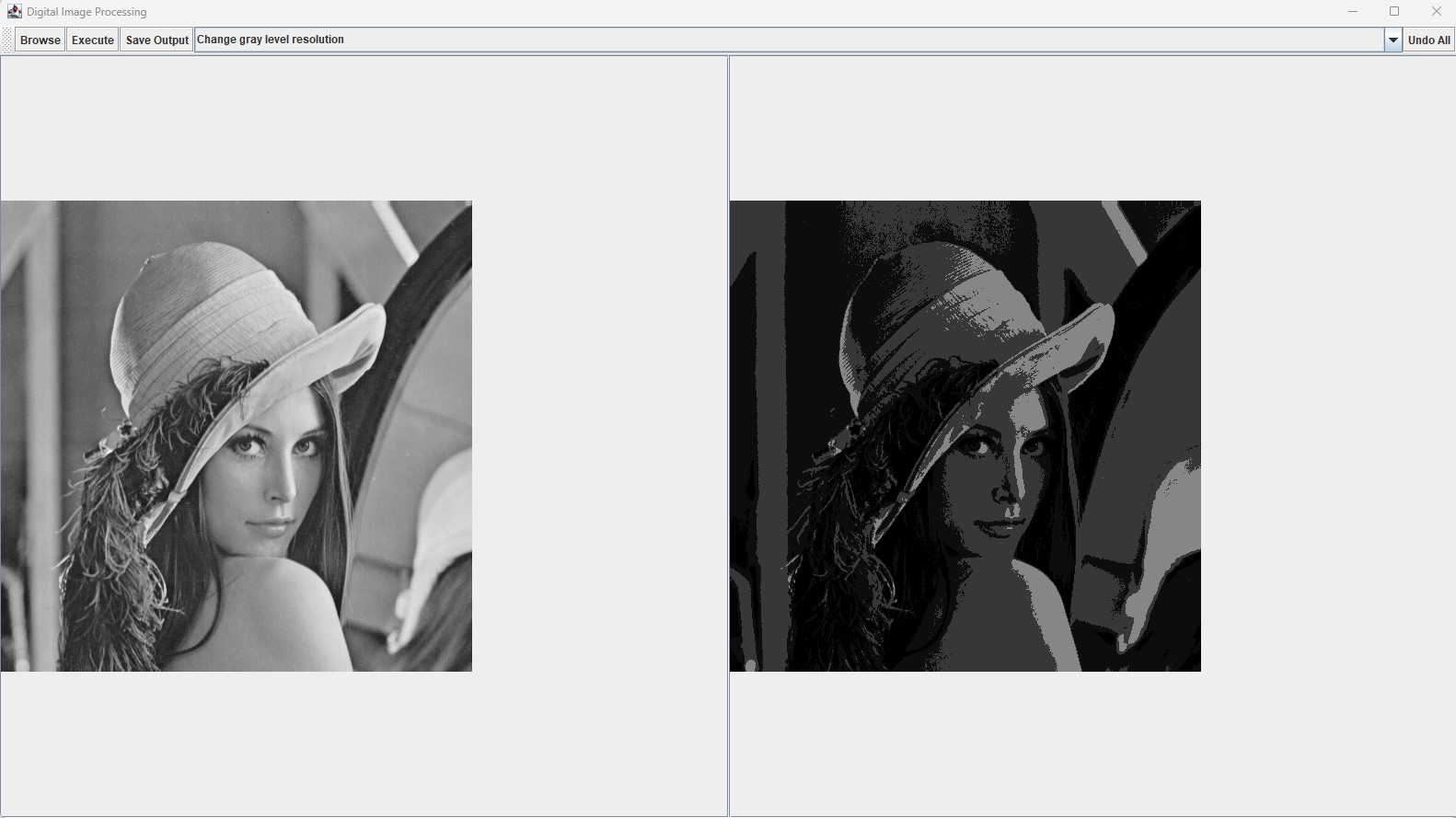
BroncoID: 015765555

Github & Source code:

https://github.com/fidelisprasetyo/DigitalImageProcessing

# Program Description

Preview of GUI of the program:



Description:

* Left image: the original image.
* Right image: the processed image.
* Browse button: to open the desired image file.
* Execute button: apply the chosen action.
* Save output: save the processed image (right image) to a file.
* Undo all: revert all changes to the original image.

Implemented Features:

* Convert to PGM: convert the loaded image file into a PGM file.
* Change spatial resolution: resize the spatial resolution of the image using:
  + Nearest neighbor method
  + Linear interpolation
  + Bilinear interpolation
* Change gray level resolution: reduce the gray level of the image.

# Program Demonstration

1. Convert to PGM

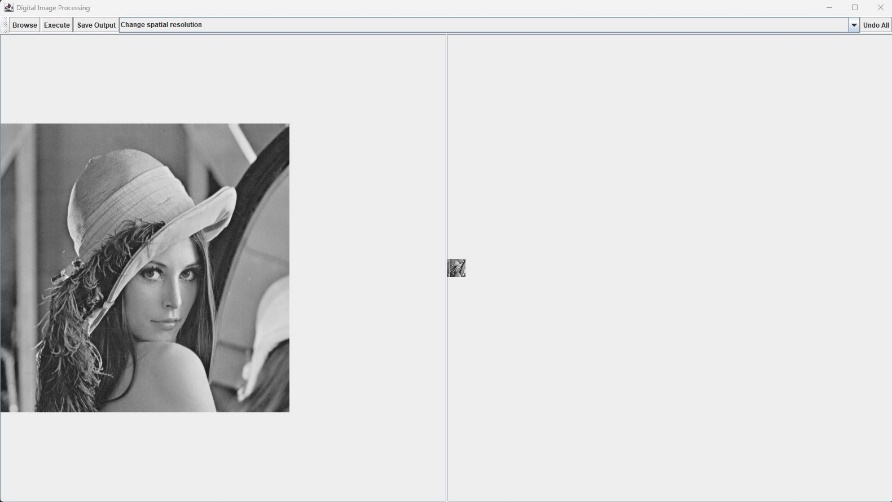
A screenshot of a computer

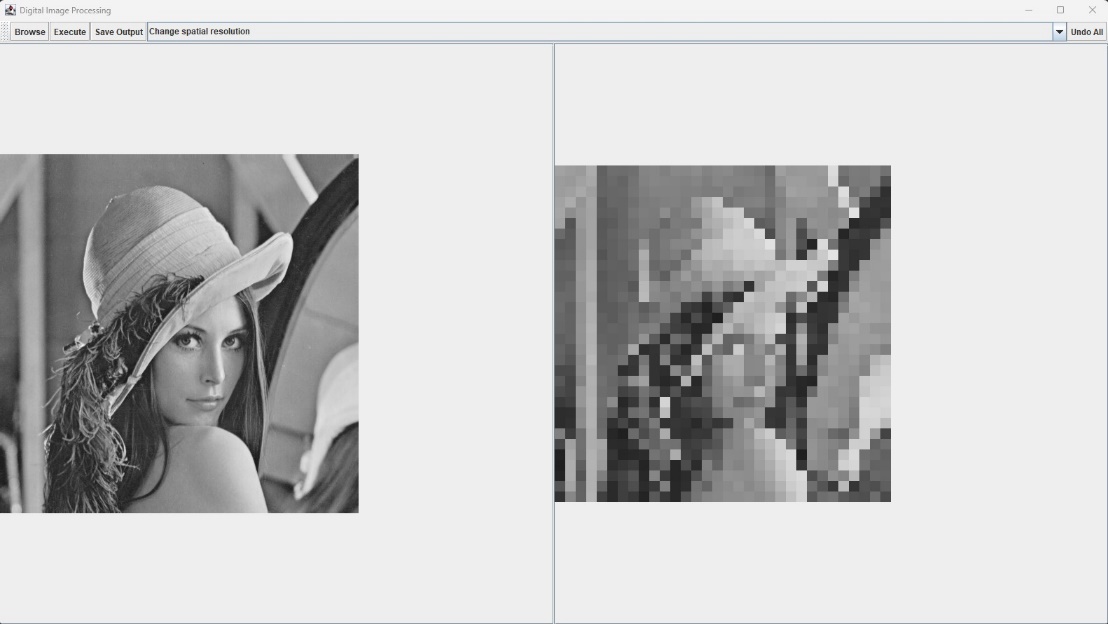
Description automatically generatedThe program is able to convert the loaded image into a readable PGM file. Here’s the preview of the content inside the pgm file of lena.gif.

1. Spatial Resolution

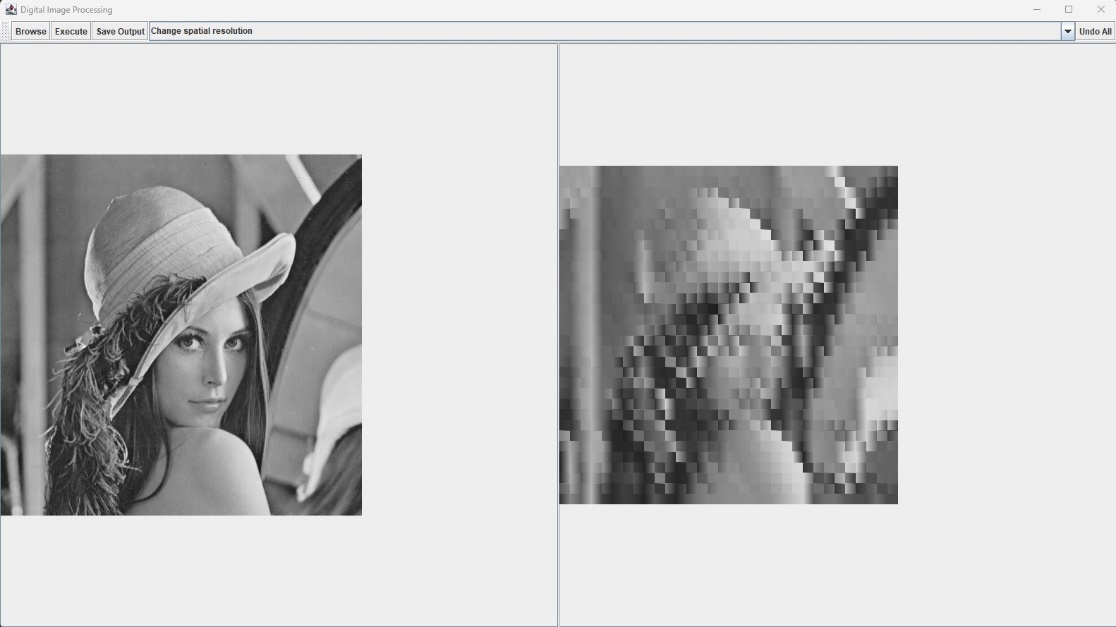
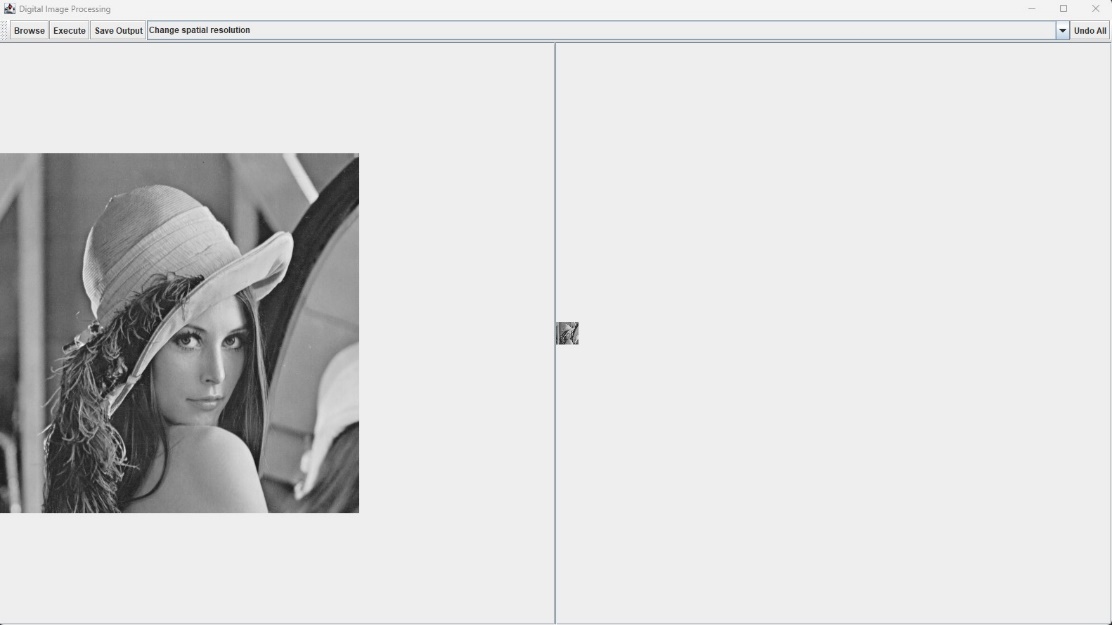
The original lena.gif image will be resized from 520x520 pixels into 32x32 in this demo. After that, the resized image will be zoomed back in into 480x480 pixels image to compare the difference between the scaling algorithms.

* Nearest Neighbor

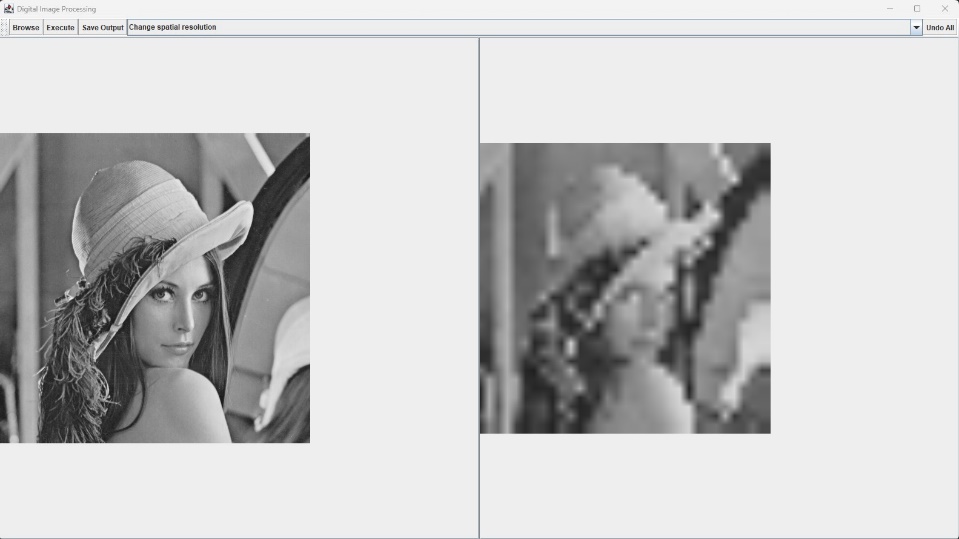
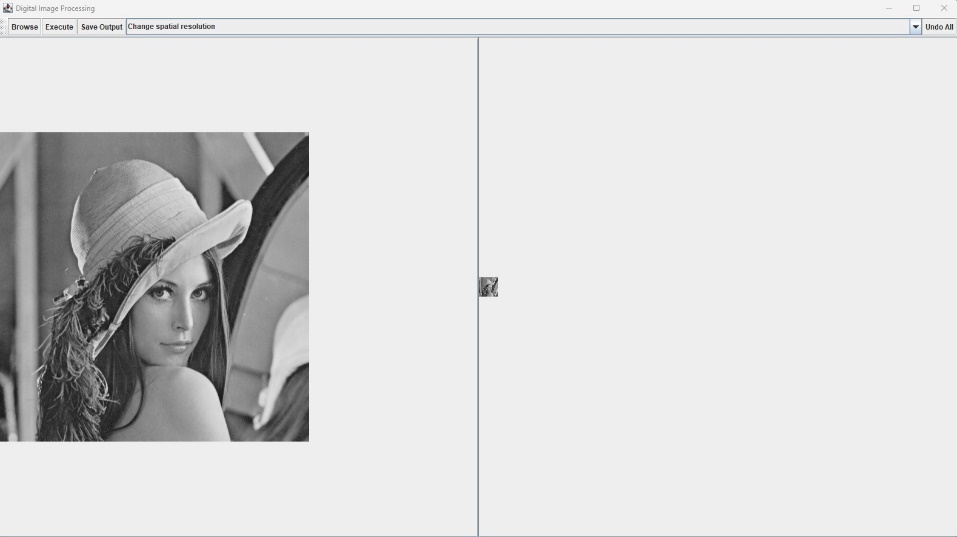




* Linear Interpolation



* Bilinear Interpolation



1. Gray Level Resolution

The original 8-bit image will be reduced to 7-bit and 2-bit images in this demo. The results, however, are not perfect as the output images are rather dark.

2-bit

7-bit

I suspect that the problem is in the thresholding. Theoretically, in a 1-bit image, we would want to categorize the 8-bit pixel values into binary values, which are 0 (black) and 255 (white). To decide the categorization of the pixel values, theoretically, we would pick the middle range of the original gray level resolution, which is 128. In other words, all pixels ranging from 0 to 127 would be converted to 0, while 128 to 255 would be converted to 255. However, the implemented feature doesn’t behave that way. I’m still experimenting with other ways to implement this feature better, but I think this current implementation would suffice for now.

# Source Code & Supporting Files

The source code, this pdf file, and output images can be obtained from this GitHub repository:

https://github.com/fidelisprasetyo/DigitalImageProcessing