Note: I am using the macro $(WXWIN) for the libraries, Tesla Computers already have the environment variable in them, and the Additional Dependencies are using WxWidget 3.1.X versions, so if there is wxmsw31u_core.lib not found error, it is probably because you are using and older version of WxWidget.

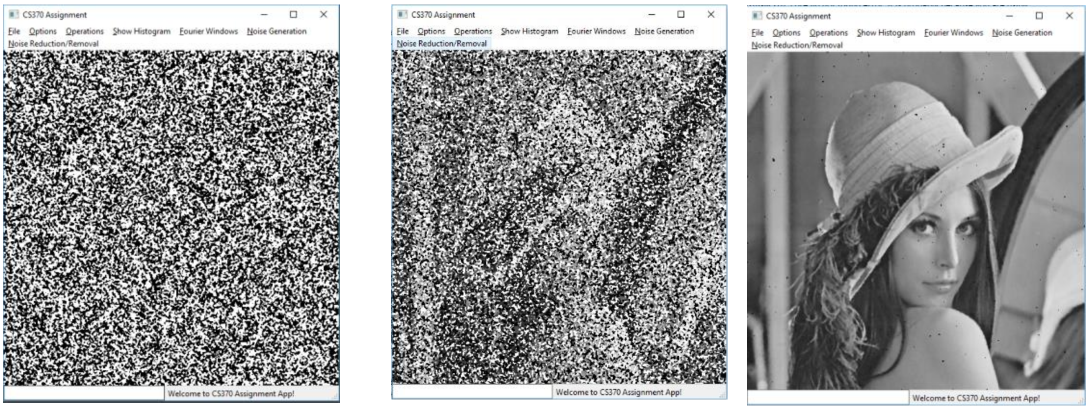
This project contains CS370.cpp, which is both the framework and the assignment for CS370 assignment 4 and the functions for the operations are located at A4.cpp.

Noise Reduction Implementation details

Under the Noise Generation are all the Gaussian and Salt and Pepper noise. And Under the Noise Reduction/Removal are Median Filter, Local Noise Reduction and Adaptive Mean Filter.

Part B: Noise Reduction / Removal

(c) Apply median filtering to the image obtained in step (b) above. Explain major differences in your results for each of the three cases (5 points)



The above image from left to right are Pa = Pb = 0.5, P(a) = 0.40, P(b) = 0.60, P(a) = 0.01, P(b) = 0.90 respectively after applying median filter.

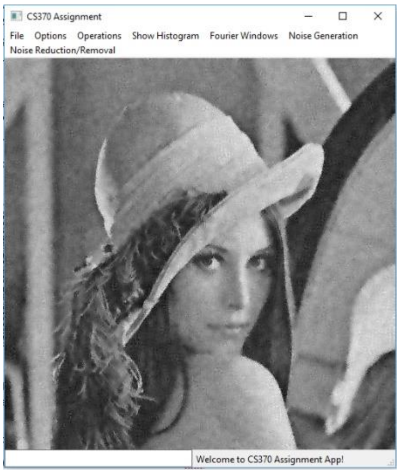
As you can see the P(a) = 0.01, P(b) = 0.90 is the clearest, as the salt and pepper noise is not so dense so when we get the medium value, we can approximate the value better.

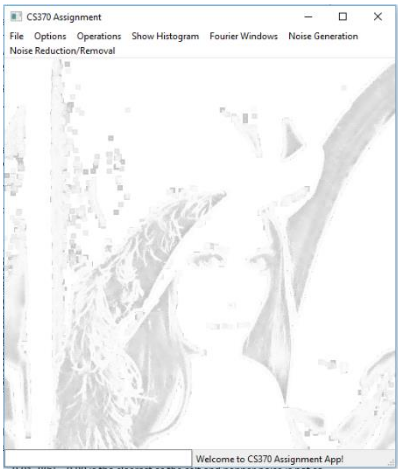
For the P(a) = 0.40, P(b) = 0.60, we can see slight outline of the lena after the median operation as the salt and pepper noise is denser so when we do a median filter, the values will be more off.

Lastly for the Pa = Pb = 0.5, we can see there is no trace of the original image at all, as the noise is so dense that it overrides the whole image, there is no trace of the original neighbour values for us to use a median filter.

Part C: Local Noise Reduction method (10 points)

For the Local Noise Reduction, to mimic the arithmetic mean filter, I set an edge case such that if you pass in a negative number. It will set all the noise variance to be the local variance making it into an arithmetic mean filter. As variance will never be negative.





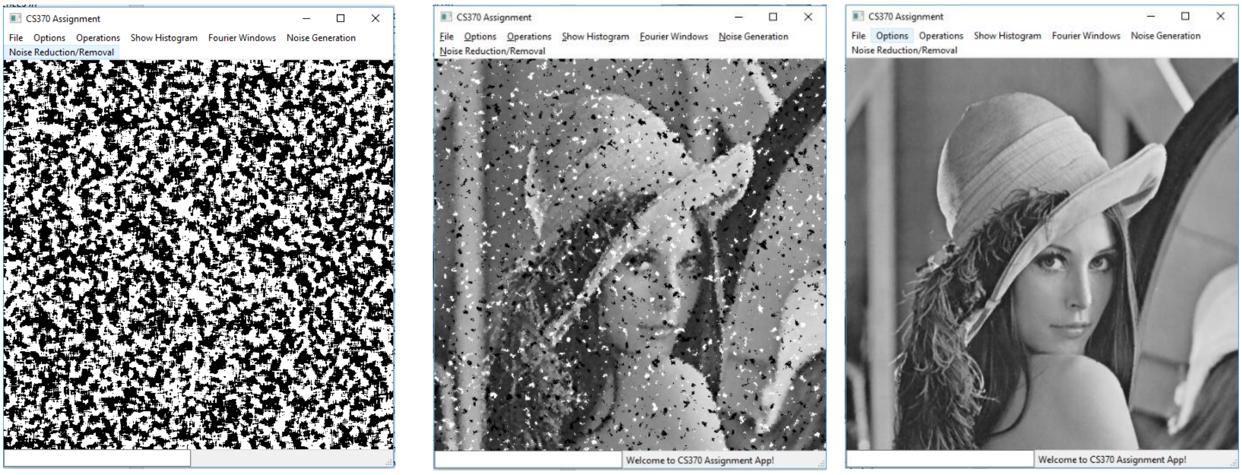
The above are the result shown for local noise reduction, where the leftmost is mean = 0, variance = 100, followed by mean = 0, variance = 1000 and mean = 172, variance = 52

The above are the result shown for arithmetic mean filter, where the leftmost is mean = 0, variance = 100, followed by mean = 0, variance = 1000 and mean = 172, variance = 52

We can see that the local noise reduction gives a sharper image, whereas the arithmetic mean filter's output is more blurred.

Part D: Adaptive Median Noise Reduction method (10 points)

The starting window size for my adaptive median noise is 3. So, it will run once in window size 3 before checking for terminating case.



The above image from left to right are Pa = Pb = 0.5, P(a) = 0.40, P(b) = 0.60, P(a) = 0.01, P(b) = 0.90 respectively after applying adaptive noise reduction with max size of 7.

As you can see the P(a) = 0.01, P(b) = 0.90 when compared with the median filter, we can see that there are no traces of noise at all.

For the P(a) = 0.40, P(b) = 0.60, for the median filter, we can only see the outline of the original image. However, for the adaptive median reduction, we can see that we can see most of the original image. There are only traces of noise remaining.

Lastly for the Pa = Pb = 0.5, similar to the median filter, we cannot see any traces of the original image as the noise is too dense. However, for adaptive median filter, we can see that the cluster of noise are merged together.

Fourier Transformation Implementation details

Under the Fourier Windows Menu, you can trigger windows which will show the result of the Fourier spectrum and the result of the reconstructed image. Triggering the window, will not affect the Fourier transformation. It will only show or hide the window.

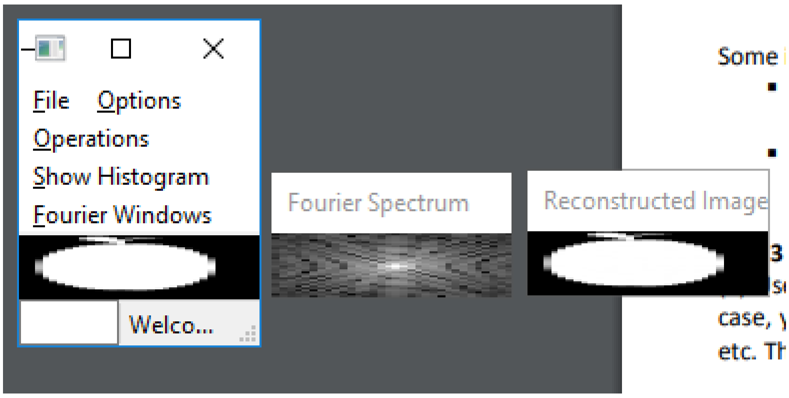
Under the Operations Menu, there is a submenu Fourier Transformations, which will generate the output of the Fourier transformations. Meaning if you hide the Fourier spectrum and the reconstructed image window, the result will still be generated.

For the Direct Fourier Transform, I used the Fourier Transform Formula directly for both the Fourier and Inverse.

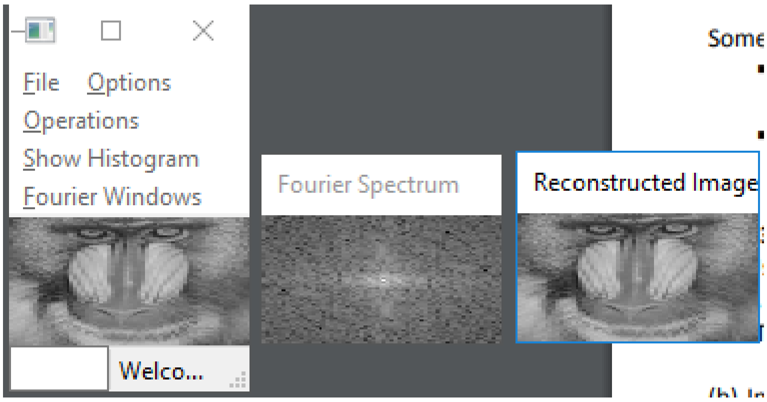
For the Separable Fourier Transform, I used the separable method for both the Fourier and Inverse, where I calculate 2 times of 1D Fourier Transform and for the Inverse I use the conjugate to calculate the reconstructed.

For the Fast Fourier Transform, I used the Fast Fourier together with the separable method, using the 2 times of the 1D FFT to calculate the result. The row then the column. For the inverse I also used 2 times of the 1D FFT and conjugate to calculate the reconstructed.

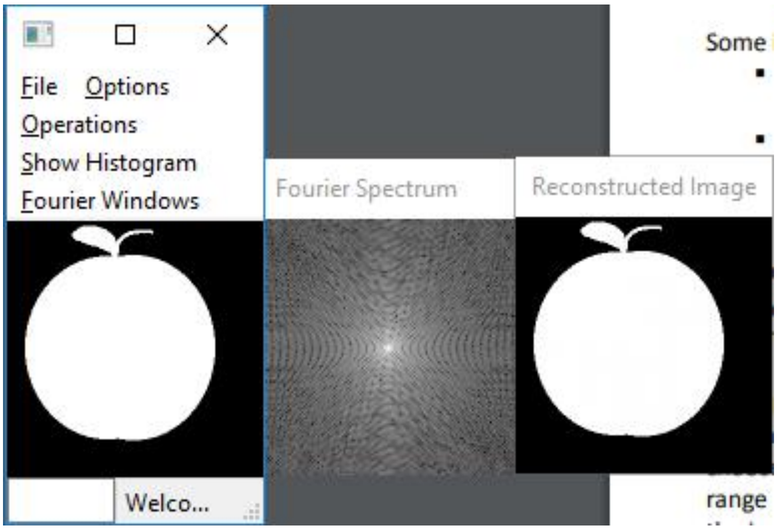
If the image of the Fast Fourier Transform is not of 2 power, I will resize it to the round downed 2 power and apply Fast Fourier to it.



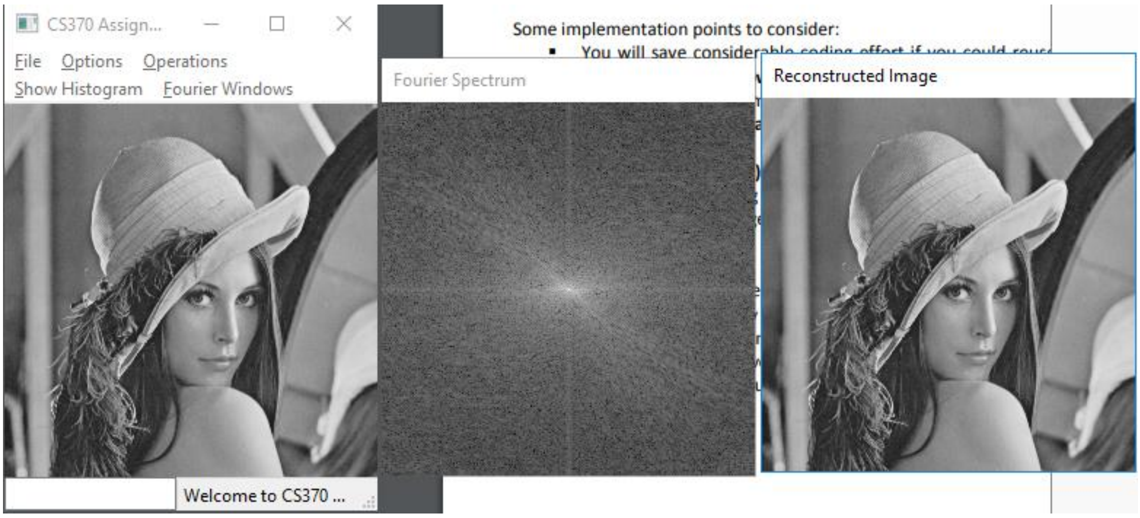
32x32



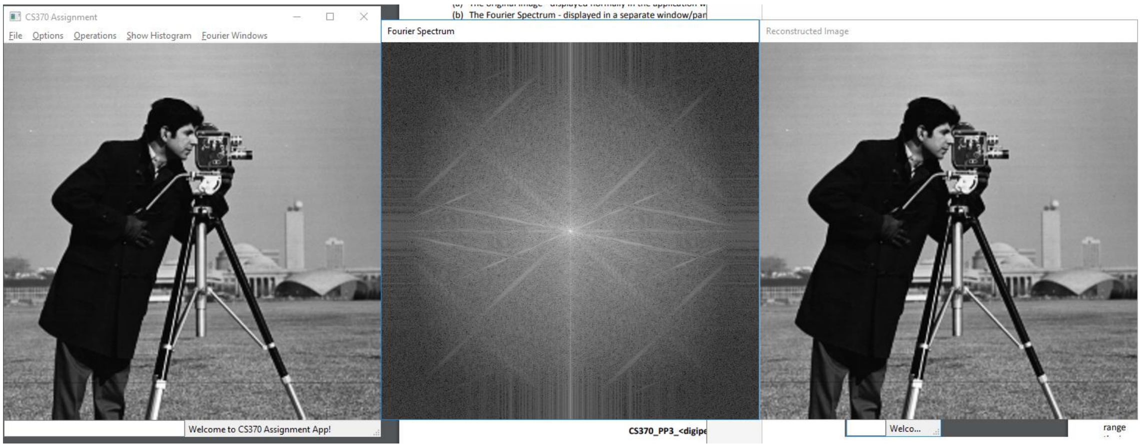
64x64



128x128



256 x256



512 x512

Runtime Timings

Type of Fourier	32x32	64x64	128x128	256x256	512x512
Direct Fourier	<0.37s	~1.44s	~19.64s	> 10min	> 30min
Separable Fourier	<0.10s	<0.34s	~1.00s	~2.97s	~22.00s
Fast Fourier	<0.005s	< 0.01s	<0.05s	< 0.10s	<0.20s

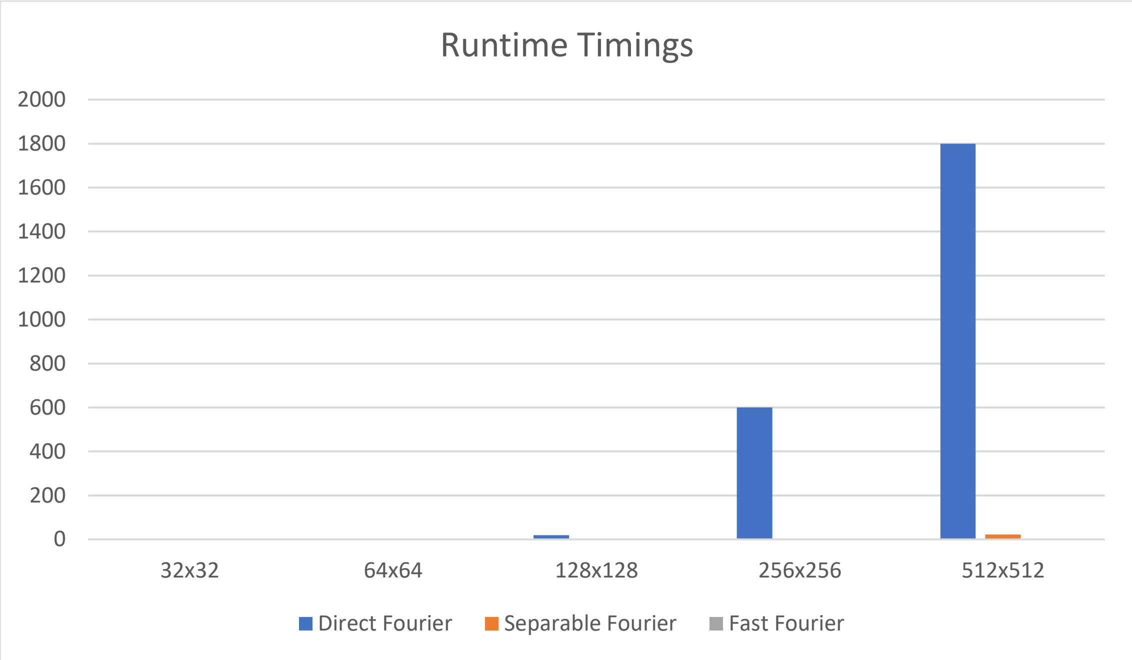


Image Operations Implementation details

Under Options, if “Bilinear Interpolation” is not selected it will use nearest neighbour interpolation and if “Bilinear Interpolation” is selected, it will use bilinear interpolation.

Under Show Histogram, if “Image Histogram” is selected, it will show the histogram under a new frame and if “Image Histogram” is not selected, it will not show it. You can trigger it to show the Histogram of the image.

For the dialog boxes in the application, please press enter for the image to preview in the dialog box.

Part A: Image Operations

For the image operations, addition, subtraction and product, I have made two file open dialog to select the two files for the operations. As stated in the handout pdf, the user will choose two files. The operation will be as followed, the first file operation latter file, e.g. if the user chooses apple-20.ppm and lena.ppm, under the operation subtraction, it will be apple-20.ppm subtract lena.ppm.

The values which exceed the image intensity for the image operations are done by clamping, however in product, I have normalized the mask image such that it will be in the range of 0 to 1.0 then multipy it to the original image.

For image negative as wxwidget has a range of 0-255, I have mapped the image during load to whatever the maximum intensity of the image is to 255 and the remaining values respectively. So, I have taken 255 as the maximum intensity, and when I save, I will map it back to the image maximum intensity as 255 and so on.

Part B: Histogram Equalization

For Histogram Equalization, I have mapped the image back to the original intensity then perform the Histogram Equalization. After the Histogram Equalization, I have mapped it back to 0 – 255 for the display.

Part C: Smoothing and Sharpening Filters

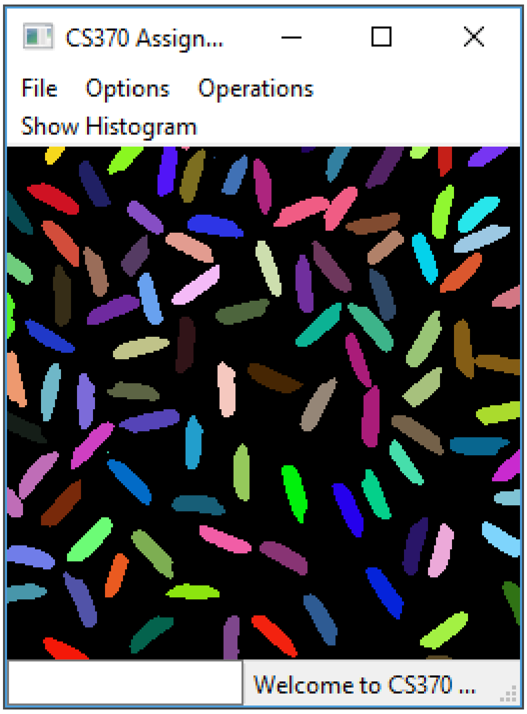
For the filters under Part C, I have taken a value repeat approach, as I feel that by using value repeat the edge case will be more accurate, as the image is like a screen shot of a scene, the border is cut off from scene, if we have the details of the surrounding pixels it will probably be somewhat similar to the edge pixel intensity value. E.g. the border of a landscape image, the sky is cut off from the border therefore, if the border is the sky and it is blue, the surrounding pixel around the border should be blue as well with a minimum difference. Thus, using value repeat will be more accurate for the border.

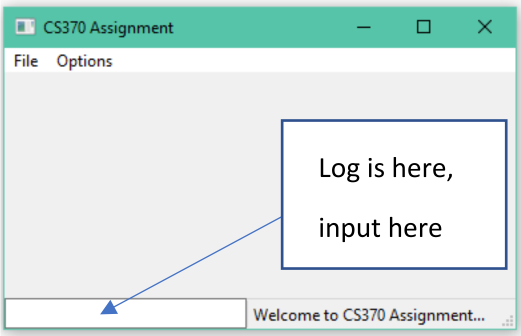
As for Unsharp-masking, I have set a default value for kernel size N and the sigma. However, if you have done Gaussian blur before, it will use those values which you have used for the Gaussian blur as stated in the handout pdf. The default N and sigma for Unsharp-mask are 3.

E.g start of application, default N = 3, default sigma = 3
    running Gaussian Blur at N = 5, sigma = 10 (default N = 5, default sigma = 10 for Unsharp)

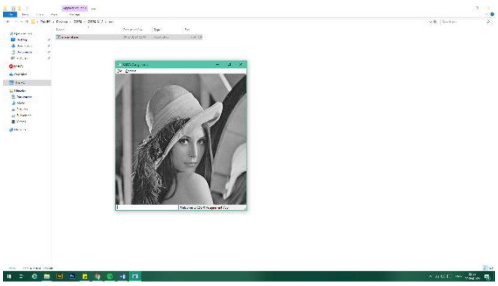
Part D: Connected Component Labeling

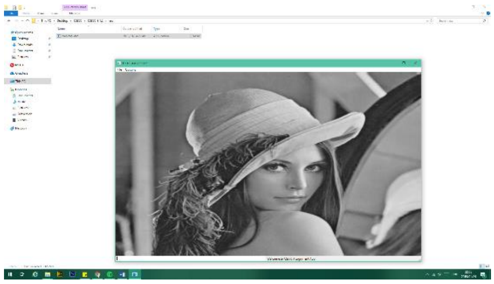
Connected Component Labeling is done in M-connected topology or M-adjacency and I have a minimum value and maximum value, to set all the values of the set. Minimum value represents the value of the minimum value in the set and maximum value represents the maximum value in the set.

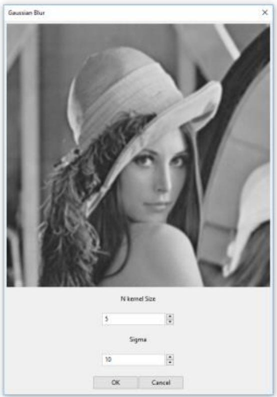


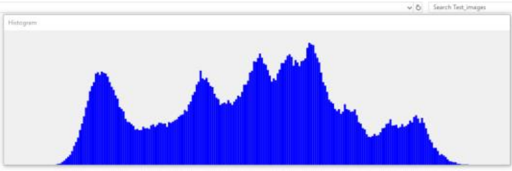


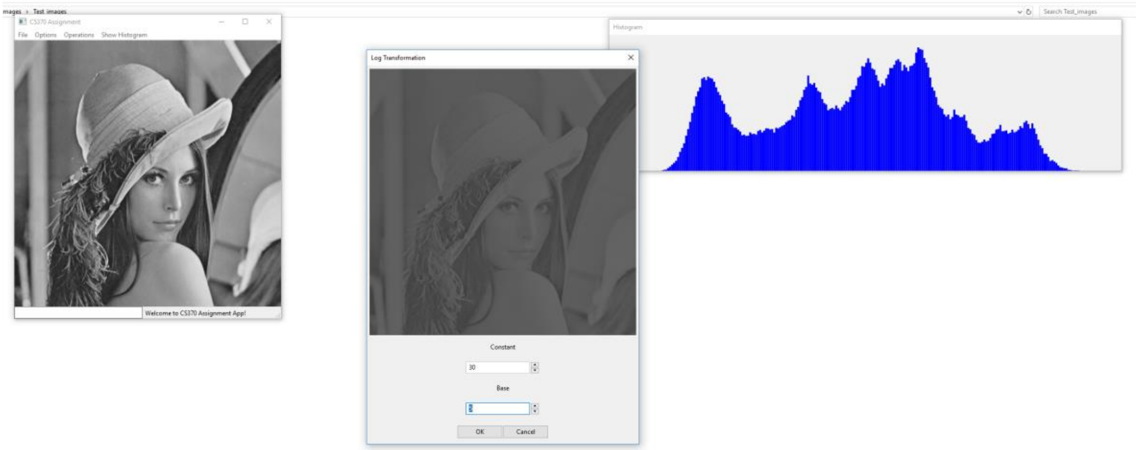
The images below are the screenshot of my executable.










images › Test_images
Search Test_images
CS370 Assignment
File Options Operations Show Histogram
Welcome to CS370 Assignment App!
Log Transformation
Constant
30
Base
OK
Cancel
Histogram