Vinit Ramesh Gore

M20CS064

Assignment 1

Question 1: Filtering

Implement a 3 x 3, 7 x 7, 9 x 9 (i) median and (ii) average filters. Apply each one of them to this image and comment on the results. Also, compare the results obtained by the two filtering methods. Do not use an inbuilt filter function.

The Barbara Image used as is:

Median Filter:

Essentially used for salt and pepper noise. The image becomes smoother as filter size increases. For 9x9 filter the edge information is visible but most of the content from the surfaces has faded. Probably because median values are replaced at every convolution point. Initial padding was kept as 128 because 0 or 255 are considered as noise for this filter.

Average Filter:

Essentially used to remove Gaussian noise. The image becomes smoother as filter size increases. For 9x9 filter the image looks blurrier probably because all points in filter window are considered while the filter ranges over the image. Due to more variations than the median filtering operation, probably the result of averaging looks blurrier. The output of all average filters is scaled to fit into the [0,255] range.

Question 2: Affine Transformation

For this image, perform the following transformations by defining the appropriate transformation matrix. (a) Translation by 2 pixels in any direction. (b) Scaling by a factor of 2 in the x-direction. (c) Rotation by 30 degrees in the anti-clockwise direction. (d) A combination of the above three operations.

[10x3+20=50]

The Barbara iamge is used as is for this question. All matrix operations are performed in Homogenous coordinate system enabling composition of matrices in the (d) part.

a) Translation by 2 pixels in +x direction:

The narrow black line at the left edge denotes the translation.

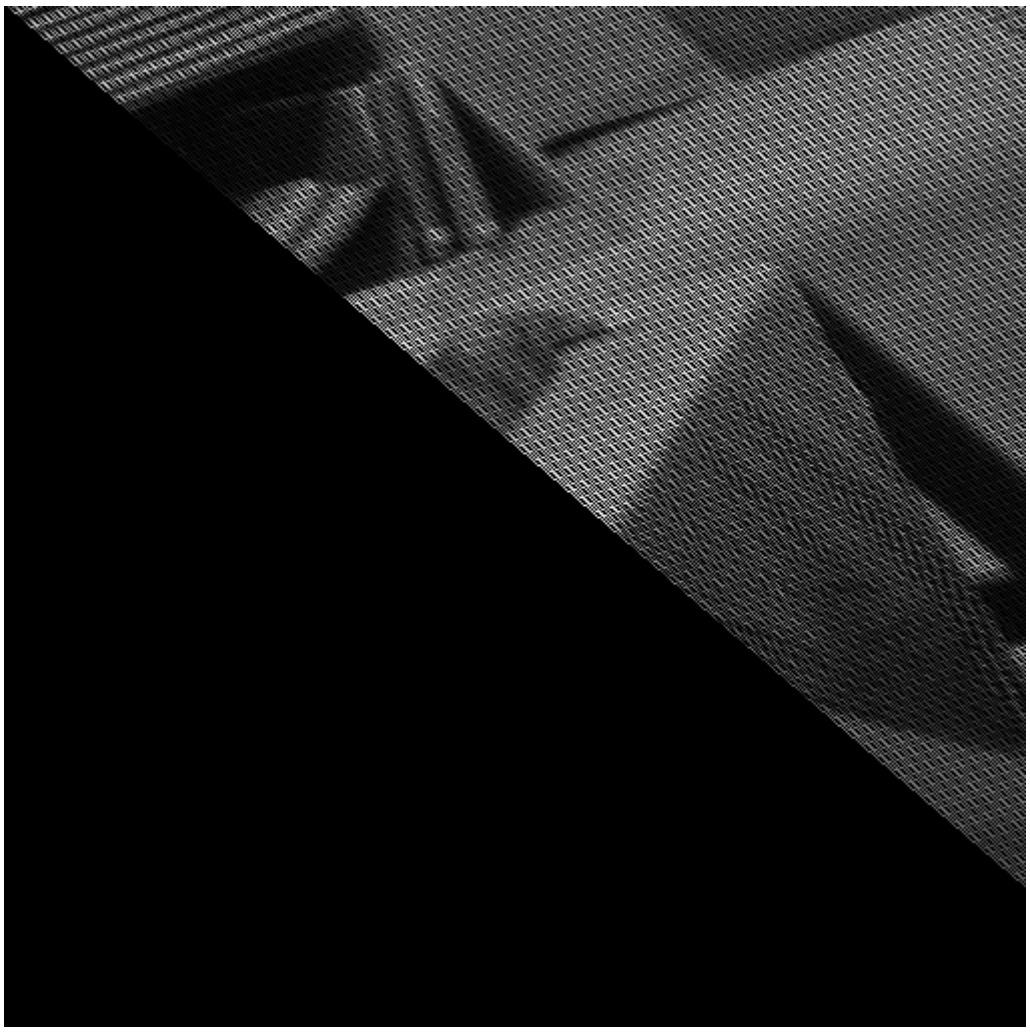
b) Scaling by a factor of 2 in the x-direction:

The pixels from the right edge got deleted as they could not fit in the frame. Nearest neighbour interpolation is used here to fill the gaps.

c) Rotation by 30 degrees in anti-clockwise direction:

Rotation operation resulted in interpolation which could not be handled with scratch code. The angle of left edge of the original image with the left edge of the result image is 30 degree.

d) Combinations of the above three operations by a single matrix:

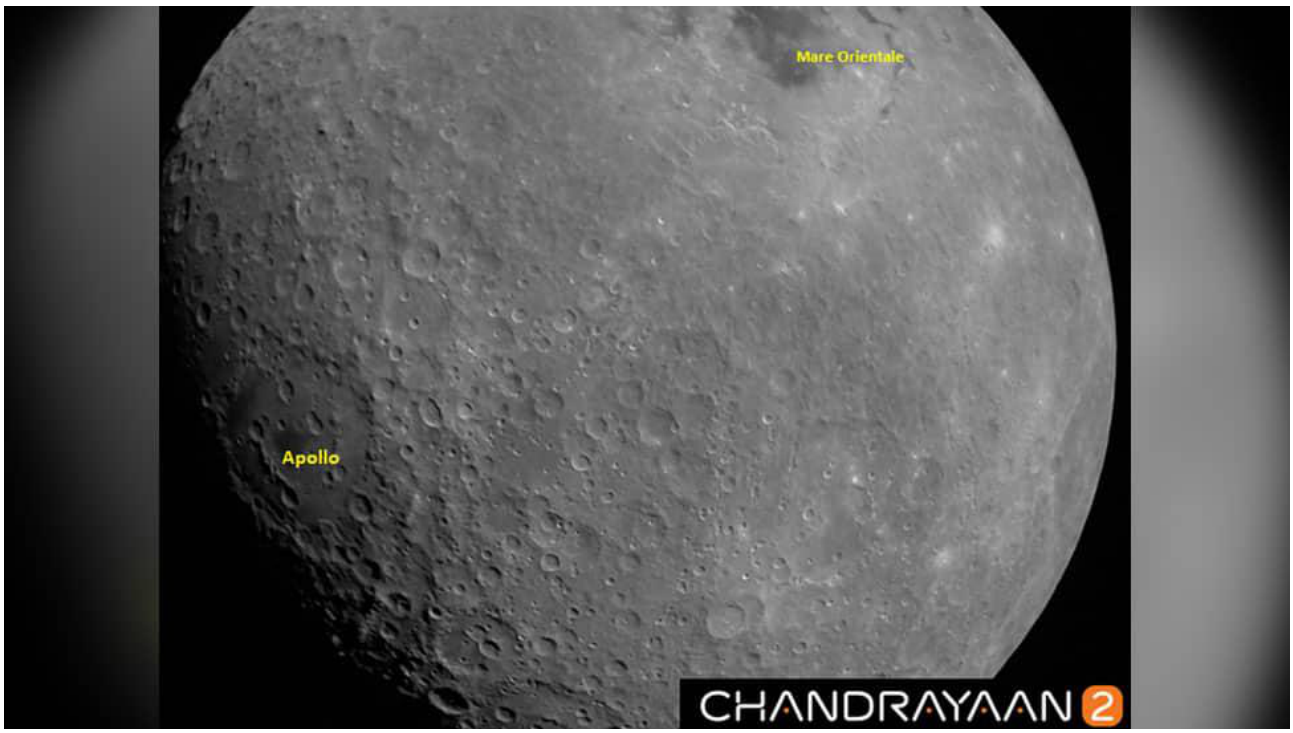


Rotation -> Scaling -> Translation was the sequence of operations.

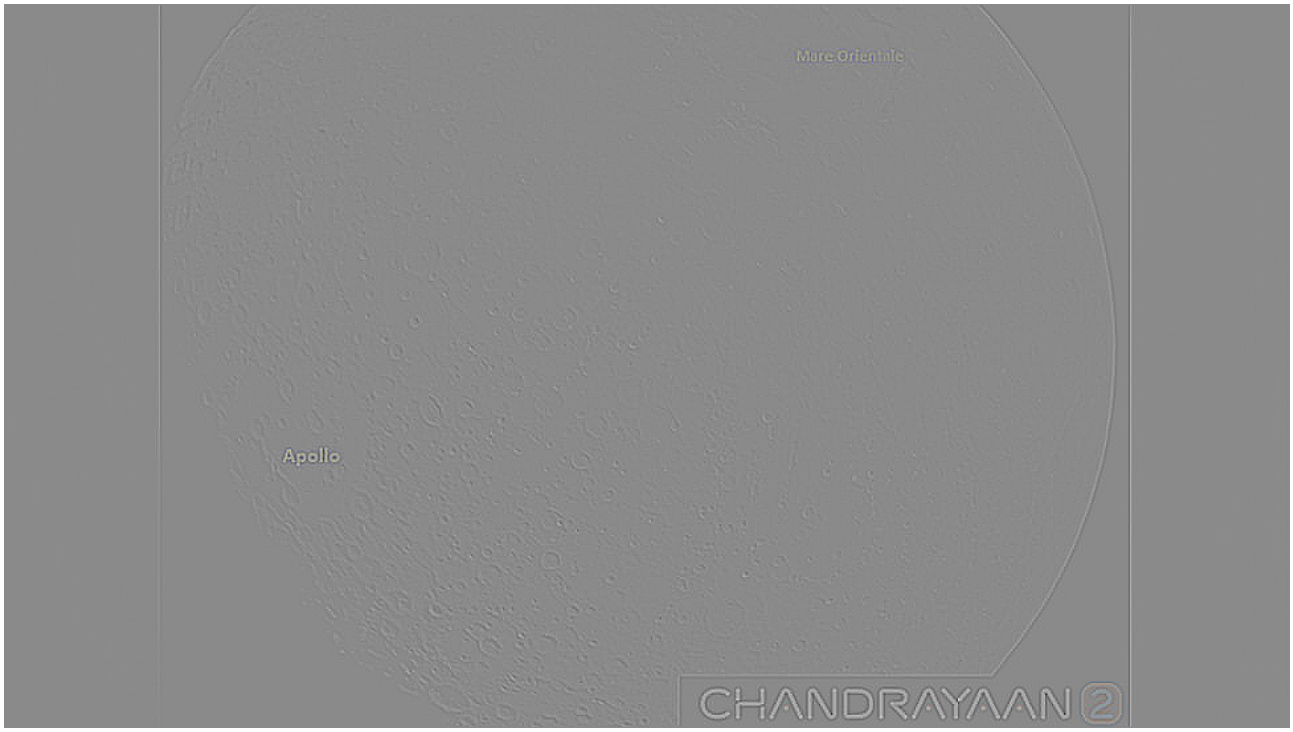
Question 3: Image sharpening through Laplacian

For this image, save the Laplacian image separately and add it to the input image to sharpen it. Now, compare it with the sharpening filter - as discussed in the class. Do not use the inbuilt function for Laplacian or sharpening filter. [30+10 =40]

The Chandrayaan Image used as is:



Result of the Laplacian Filter:



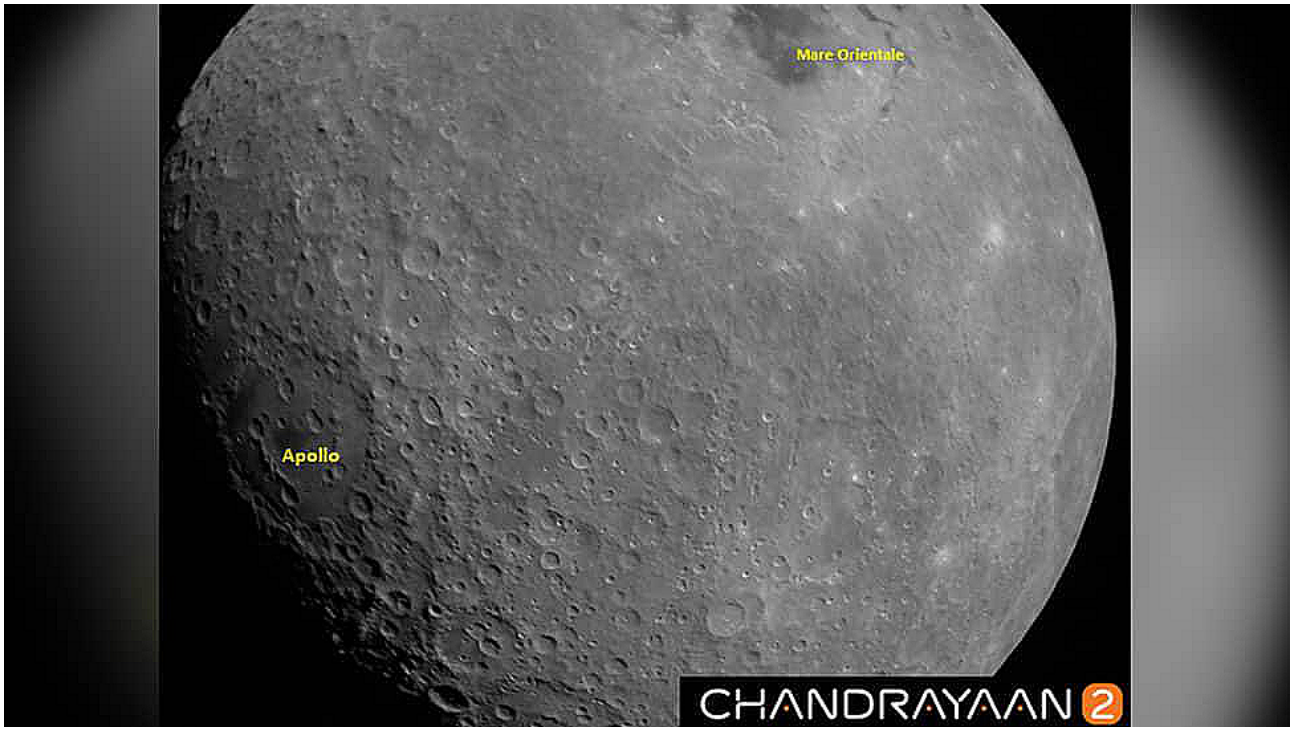
Edges are captured by this filter. The actual filter that was used had negative values also. In order to display it as is, the negative values needed to be clipped. So the scaled result is shown above. Therefore the majority of pixels in the image appear gray.

The scaling was not performed as mentioned in Gonzalves (MinMax scaling) because the result had increased intensities which made it look poor. So the min value was not subtracted from the prescaled result of convolution with filter and so the sharpened image (below) has same brightness as the original image.

The filter used is as follows:

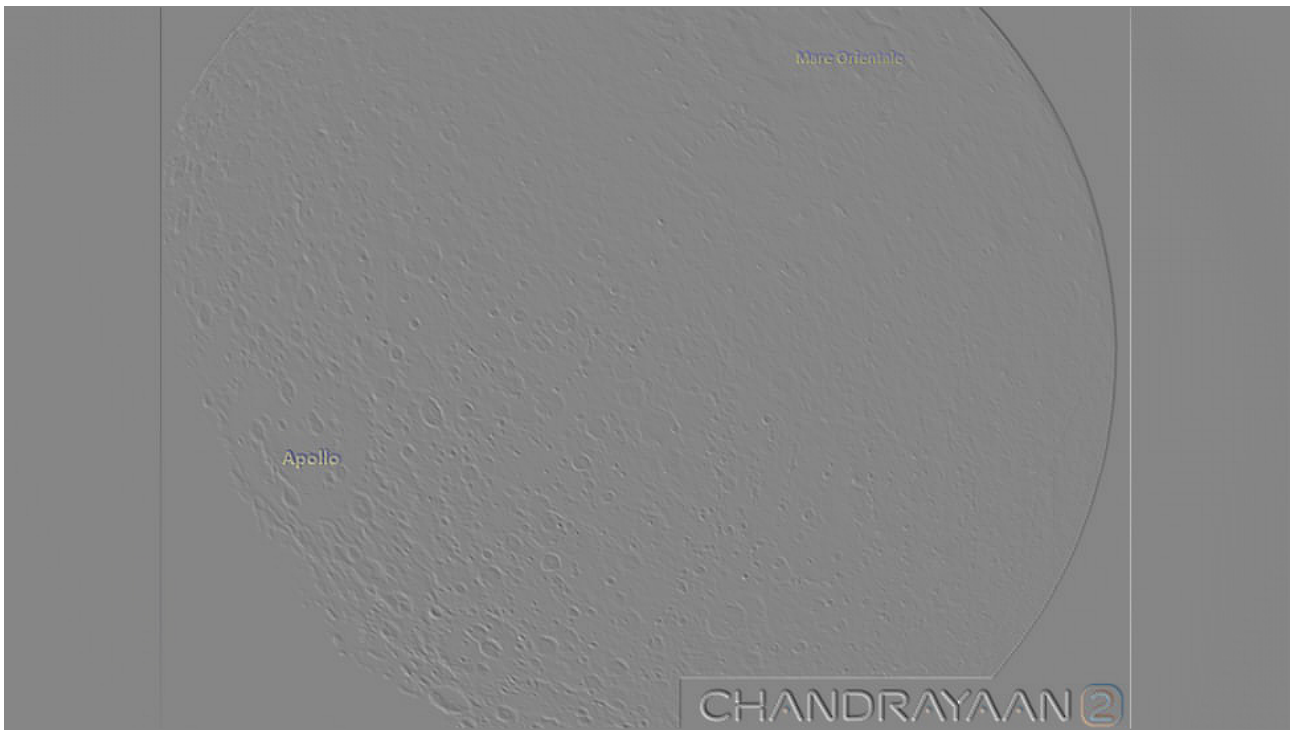
0	-1	0
-1	4	-1
0	-1	0

Sharpened Image using Laplacian:



The prescaled version of the previous result was added to the original image to get this output. The output again was scaled between 0 to 255 using MinMax scaling.

Result of First Derivative (Sharpening filter):



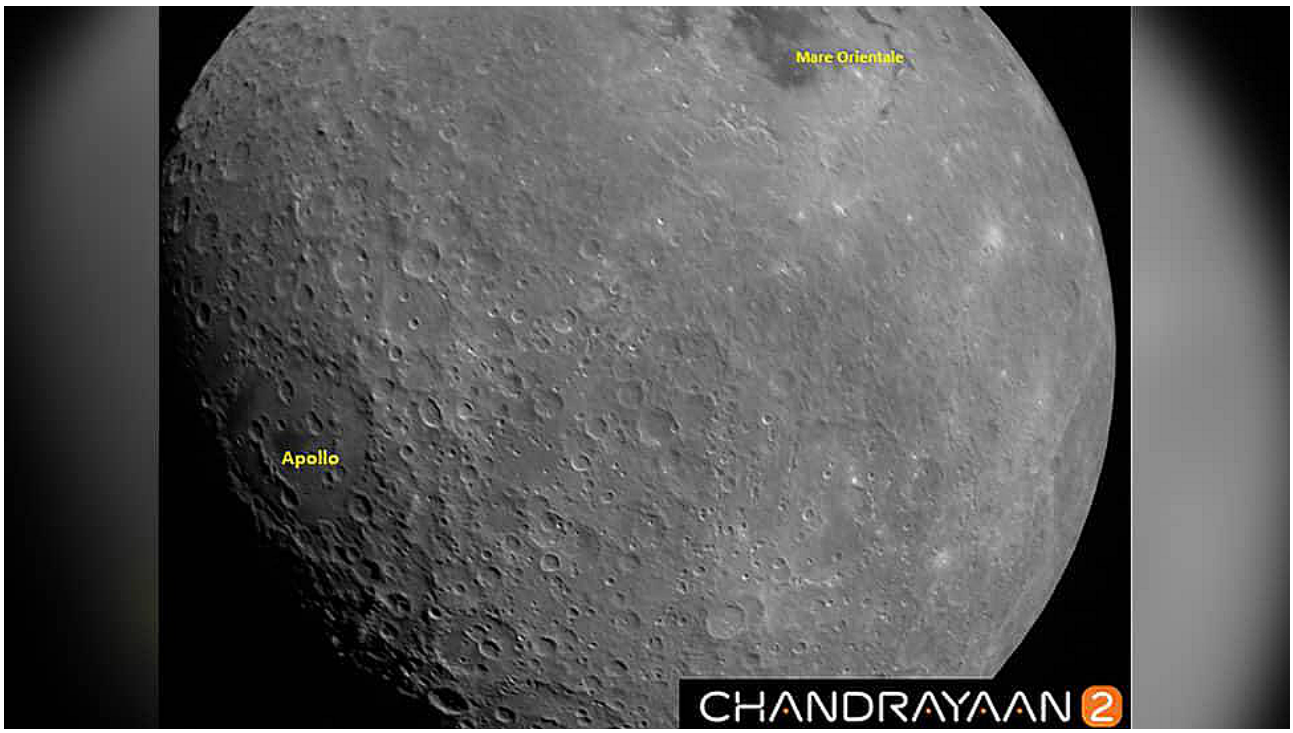
Scaled version of the result of convolving sharpening filter with the image.

The filter used is as follows:

0	-1	0
0	2	-1
0	0	0

Here also min value was not subtracted from the prescaled version. This result has less details as compared to Laplacian.

Sharpened Image using Sharpening Filter:



Output when the previous result added to the original image. Less sharper than the Laplacian operation.

Question 4: Discrete Wavelet Transform (DWT)

a) Perform DWT on this image using Haar and db 9/7 (individually) b) Perform smoothing operation on each component c) Perform Inverse DWT and obtain the smoothed image d) On the original image, apply Gaussian smoothing filter (you can use an inbuilt function here) Report all the DWT coefficients for both DWT approaches. Draw your analysis of the results of the two approaches. Is there any difference between the outputs of (c) and (d)? Justify your answer. What is the metric used to evaluate the difference between them? You are allowed to use inbuilt functions. [10+10+10+5+20+5=60]

The Cameraman Image used as is:

(a b) a: Original image, b: Result of Gaussian Blur, c-e: Result of

(c d e) smoothing in Wavelet domain using c: Haar d: db7 e: db9

The result of DWT is as good as the result of Gaussian. Infact, the relevant information is more contained in the results of DWT. Comparing haar with db9, we see less jaggies in db9 result than in the result of Haar. Here, I had later realized that the significance of axes parameter to get the desired output from Wavelet transform.

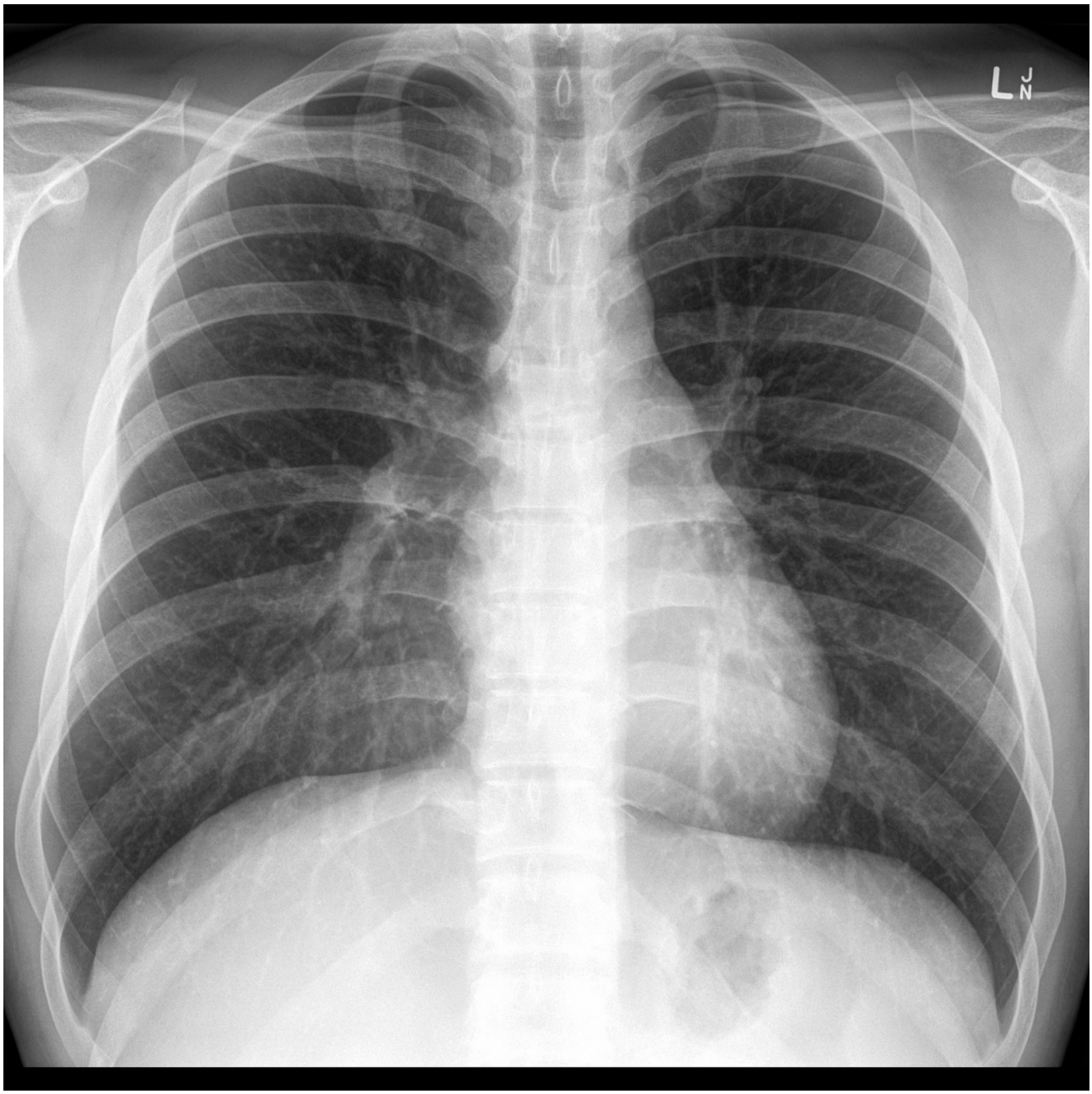
Question 5: Image Watermarking techniques

For this image, perform the invisible watermarking using any technique. Use this image as a watermark. You are allowed to use inbuilt functions. Bonus: Students will get extra marks if done from scratch.

[20] + bonus=10

The X-ray Image:

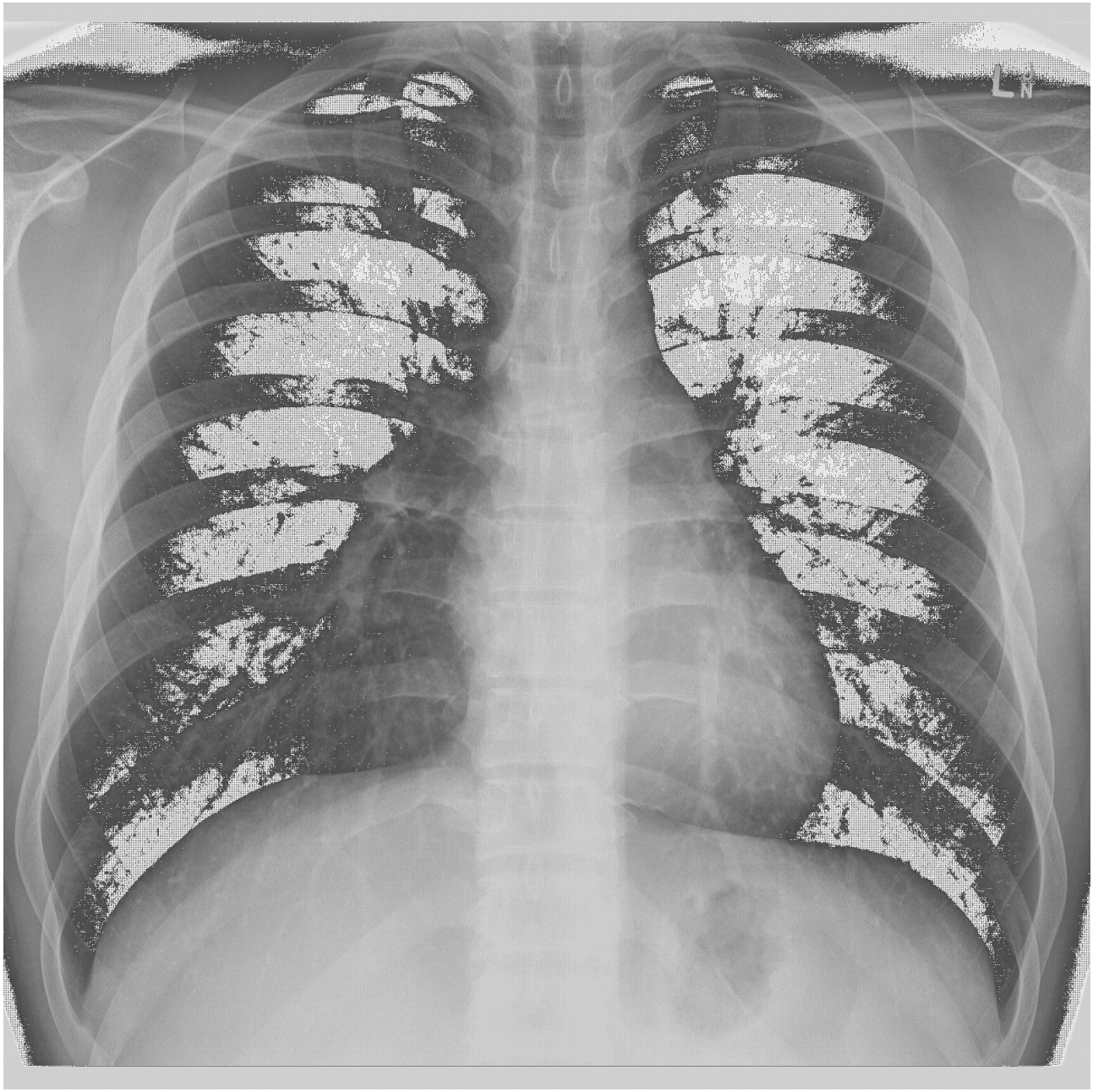
Converted to RGBA to operate with the RGBA watermark image.



The Watermark used as is:

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Result of Watermarking:



The process followed is as follows:

1. Find single level 2D Wavelet Transform of the first two axes of the xray image and the 2D Wavelet Transform with level = 8 for logo.
2. Normalize the LL component and apply thresholding to get its binary version with pixel values as 0 and 1.
3. Add the result from 2 to high frequency components of xray.
4. Find the IDWT with approximation sub-band same as that of xray and new detailed sub-bands obtained from 3.

References

- https://gist.github.com/mjdietzx/545fa2874b2688e9bcb71e2ee92cd5a0
- https://www.geeksforgeeks.org/how-to-convert-images-to-numpy-array/
- https://code.tutsplus.com/tutorials/image-filtering-in-python--cms-29202
- https://sites.google.com/iitj.ac.in/cv2020/lecture-slides?authuser=0
- https://numpy.org/doc/stable/reference/index.html
- https://stackoverflow.com/questions/41500637/how-to-extract-r-g-b-values-with-numpy-into-seperate-arrays/41500767
- https://stackoverflow.com/questions/39619222/laplacian-sharpening-grey-image-as-result
- https://www.fil.ion.ucl.ac.uk/spm/doc/books/hbf1/Ch2.pdf
- https://stackabuse.com/affine-image-transformations-in-python-with-numpy-pillow-and-opencv/
- http://www.socouldanyone.com/2013/03/converting-grayscale-to-rgb-with-numpy.html