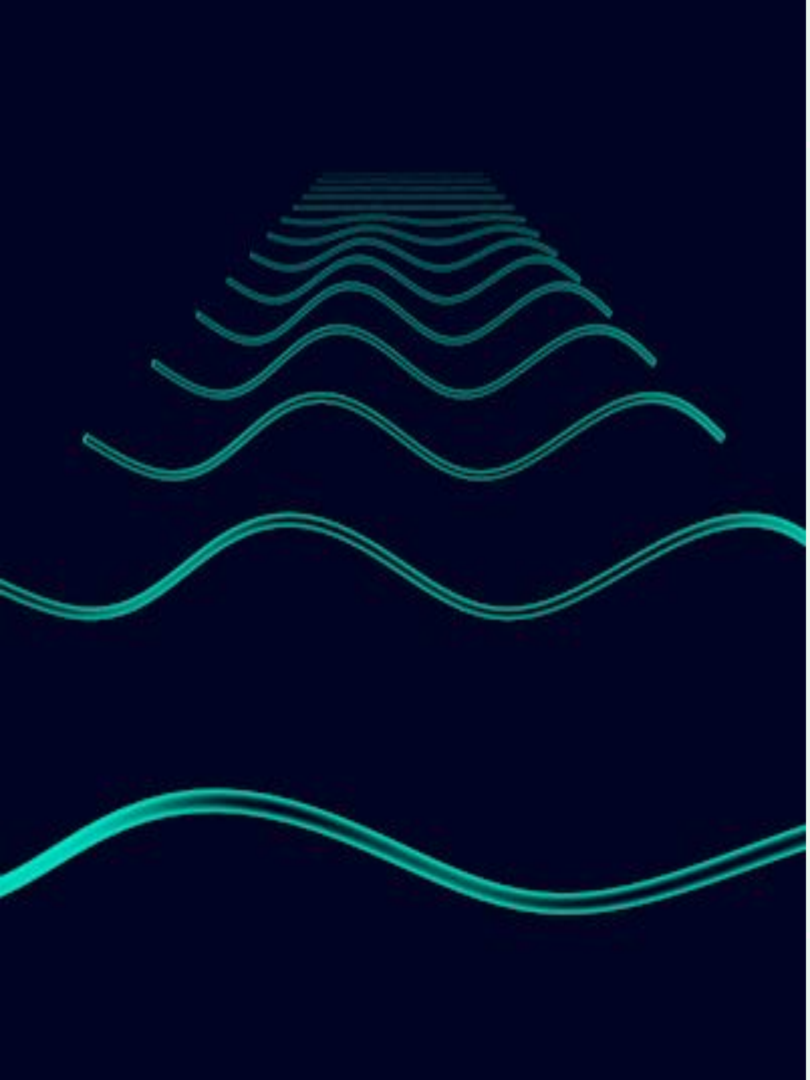


Wavelets

By:

Sumaita Binte Shorif

OUTLINE

01 Waves and Wavelets

Definitions and Comparison

02 Wavelet Transform

Shifts and Dilations,
The Haar Wavelet

03 2D Wavelets

Standard Decomposition

04 Wavelets and Images

Haar and Daubechies Wavelets
Comparison

OUTLINE (Cont.)

05 Image Compression

Image Compression, Thresholding and Quantization along with Compression by DWT Extraction

06 High Pass Filtering

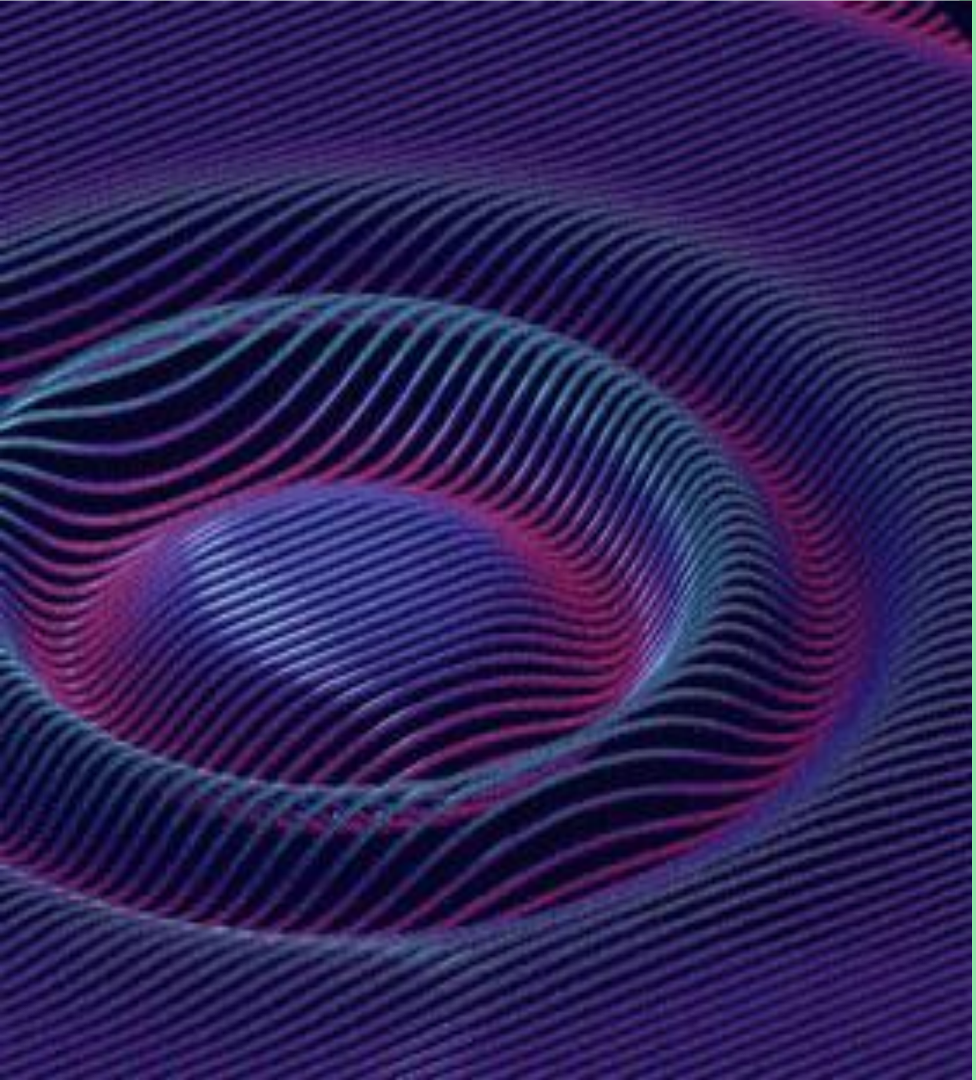
High Pass Filtering using Wavelets

07 Denoising

Denoising using Wavelets

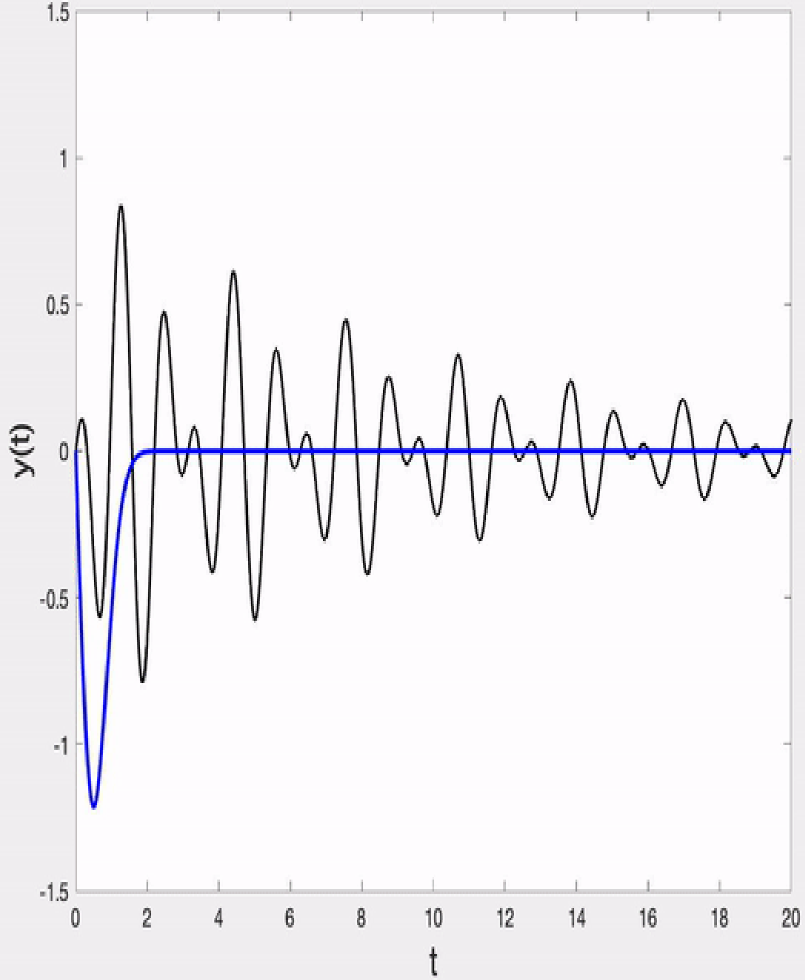
08 Conclusion

To conclude, a summary to all that we have discussed



01. Waves and Wavelets

Definitions and Comparison



Waves and Wavelets

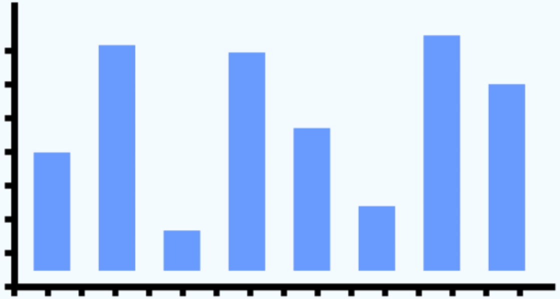
The idea of wavelets is to keep the wave idea, but drop the periodicity. So we may consider a wavelet to be a little part of a wave: a wave which is only non-zero in a small region.

Waves and Wavelets (Cont.)

- **Waves** represent the spatial distribution of pixel values and enable frequency analysis through Fourier Transforms.

- **Wavelets** are used for:

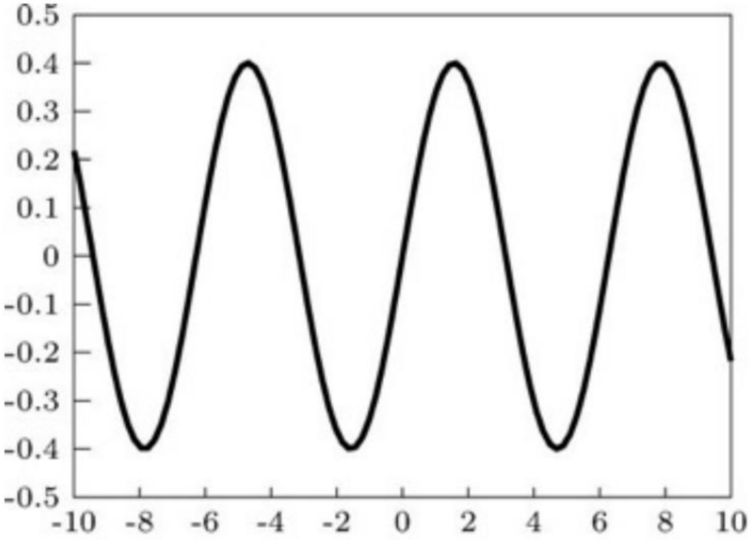
  - **Compression**
  - **Denoising.**
  - **Feature Extraction**
  - **Image Enhancement**
  - **Multiresolution Analysis Image Fusion**



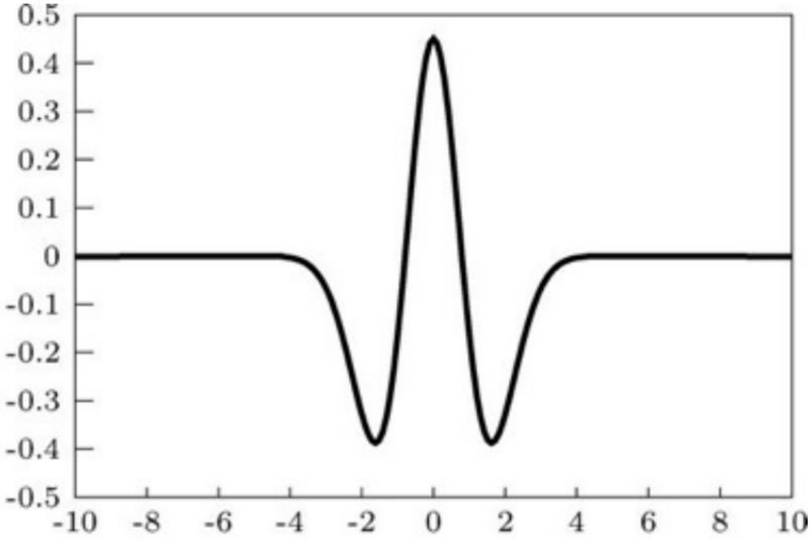
- Wavelets excel in capturing localized features and are vital tools for advanced image manipulation and analysis.

Waves and Wavelets (Cont.)

Comparing a Wave and Wavelet



(a) A wave

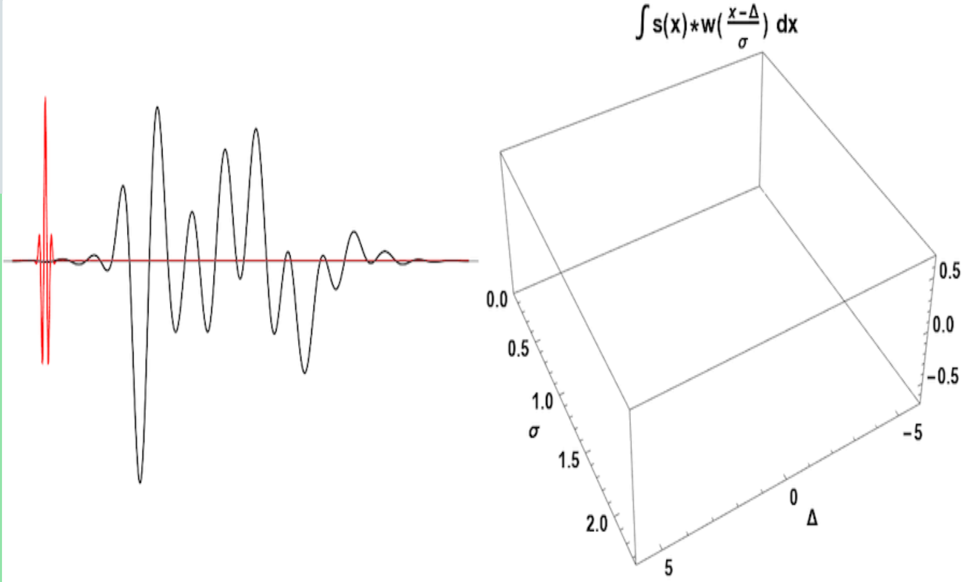


(b) A wavelet

02.

WAVELET TRANSFORM

Shifts and Dilations,
The Haar Wavelet



Wavelet Transform

All wavelet transforms work by taking weighted averages of input values and providing any other necessary information to be able to recover the original input

The Haar Wavelet

The Haar wavelet has been around for a long time, and has been used with images as the Haar transform. Only recently has the Haar transform been viewed as a simple wavelet transform. In fact, the Haar wavelet is the simplest possible wavelet,

The Haar Wavelet

Python Code

```python
# Creating a Wavelet object for Haar wavelet
haar_wavelet = pywt.Wavelet('haar')

# Getting the wavelet function values
wavelet_values = haar_wavelet.wavefun()

# Extracting the wavelet function components
scaling_function, wavelet_function, x_values = wavelet_values

# Plotting the scaling function and wavelet function
plt.figure(figsize=(10, 6))
```

Fig: Python code for Haar Wavelet

The Haar Wavelet

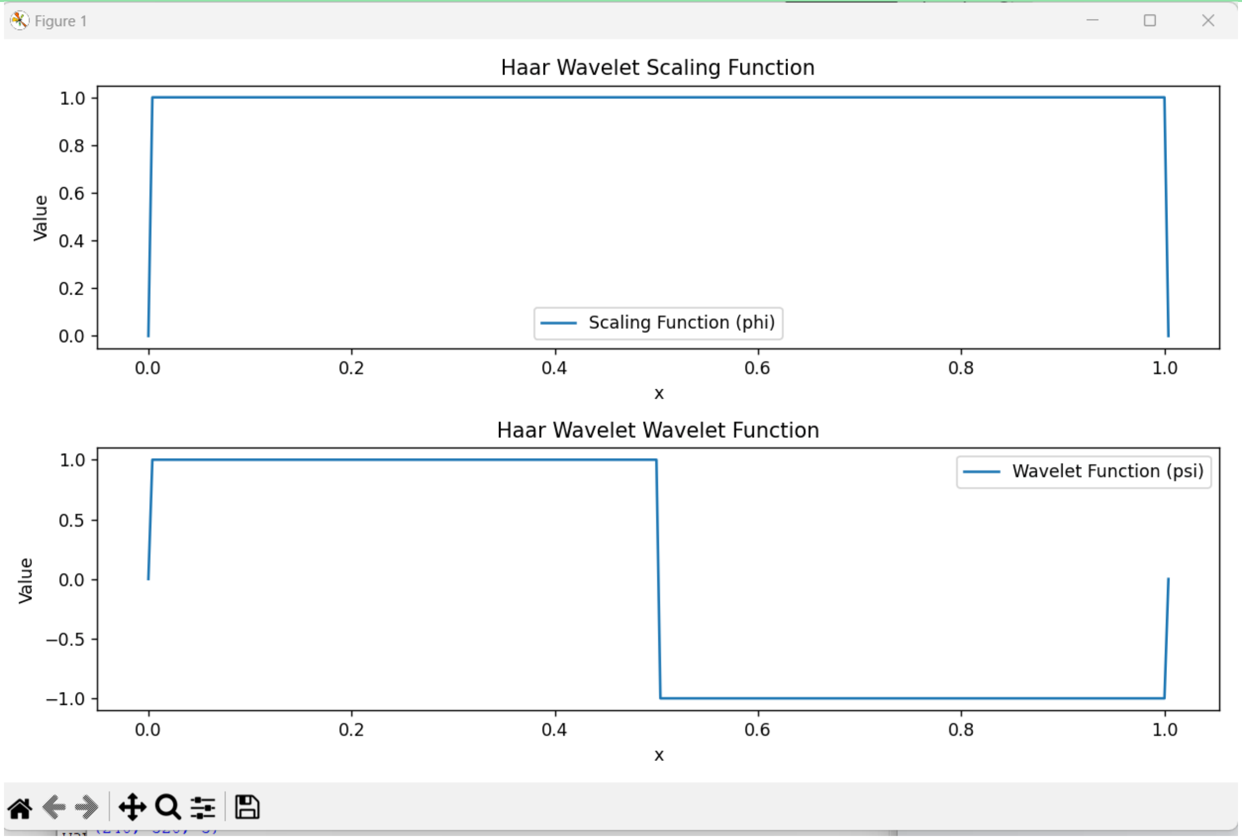


Fig: Output Obtained for Haar Wavelet

Wavelet Transform (Cont.)

Any wavelet transform can be computed by matrix multiplications. A wavelet transform contains within it both high and low pass filtering of our input, and we can consider a wavelet transform entirely in terms of filters.

Wavelet Transform (Cont.)

```
c = pywt.data.camera()

coeffs2 = pywt.dwt2(c,'db2')

cA,(cH,cV,cD)= coeffs2

plt.subplot(2,2,1)
plt.axis('off')
plt.title('Approximate image LL')
plt.imshow(cA, cmap='gray')

plt.subplot(2,2,2)
plt.axis('off')
plt.title('Horizontal detail image LH')
plt.imshow(cH, cmap='gray')


plt.subplot(2,2,3)
plt.axis('off')
plt.title('Verical detail image HL')
plt.imshow(cV, cmap='gray')

plt.subplot(2,2,4)
plt.axis('off')
plt.title('Diagonal detail image HH')
plt.imshow(cD, cmap='gray')

plt.show()
```

Python Code for DWT

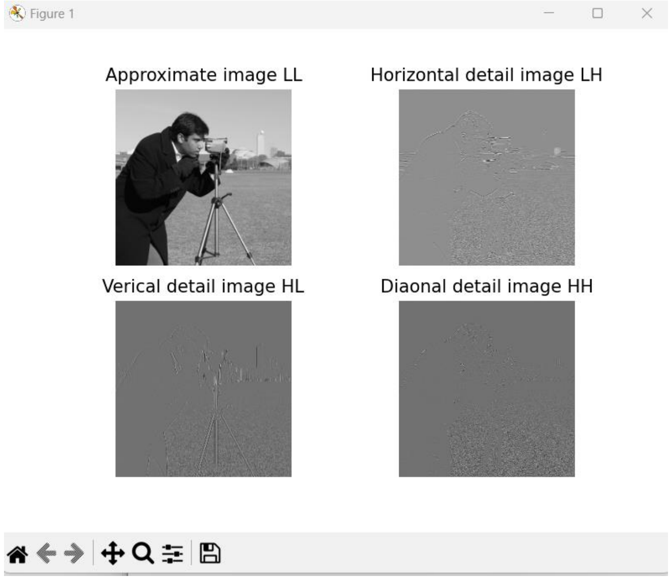
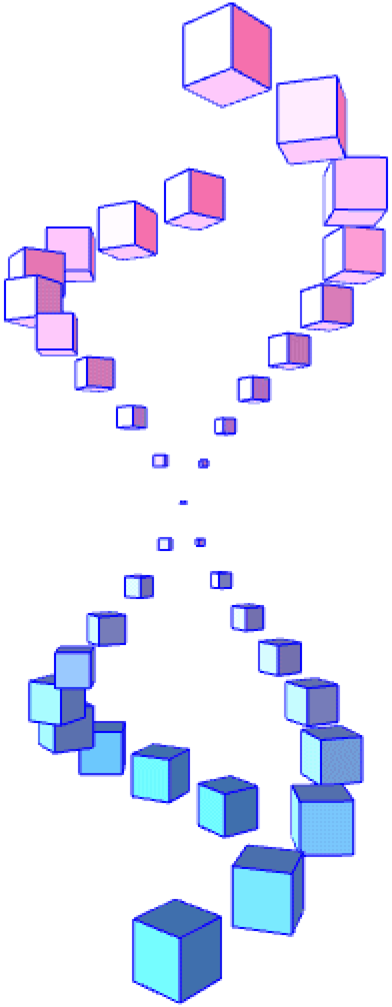


Fig: Output Obtained for Discrete Wavelet Transform

03.

2D Wavelets

Standard Decomposition

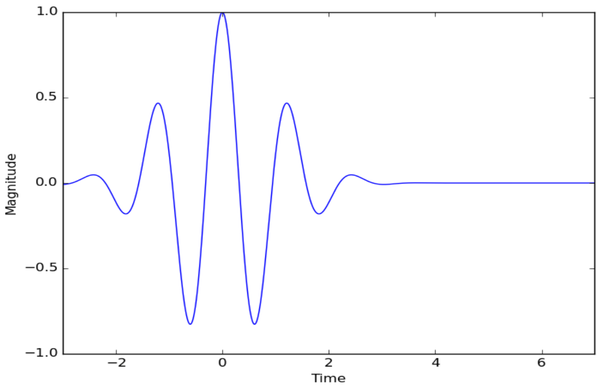


Two Dimensional Wavelets

- The two-dimensional wavelet transform is separable, which means we can apply a one-dimensional wavelet transform to an image in the same way as for the DFT:

- we apply a one-dimensional DWT to all the columns, and then one-dimensional DWTs to all the rows of the result. This is called the standard decomposition

2D Wavelets (Cont.)



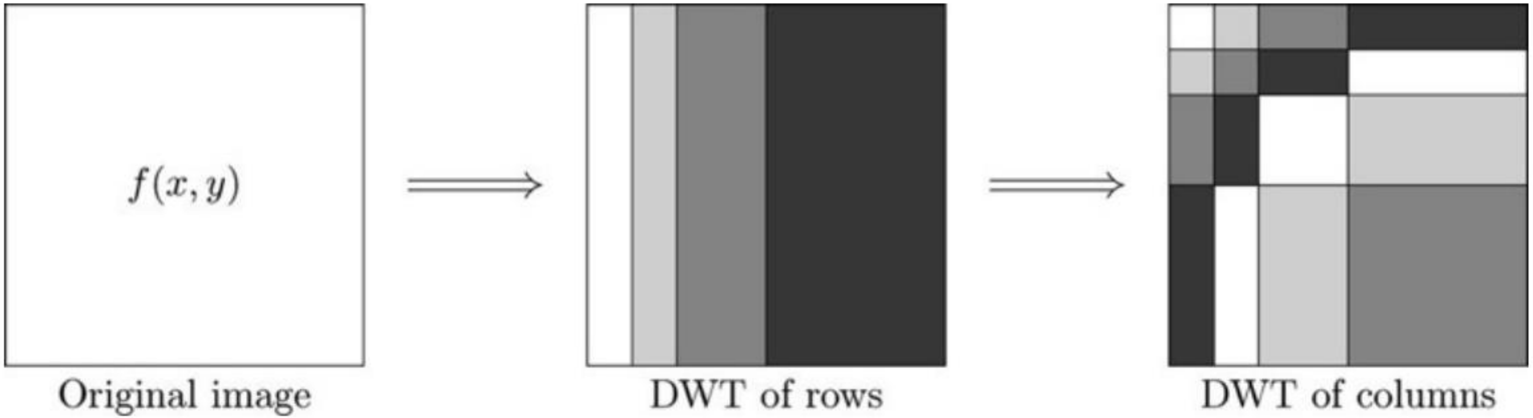
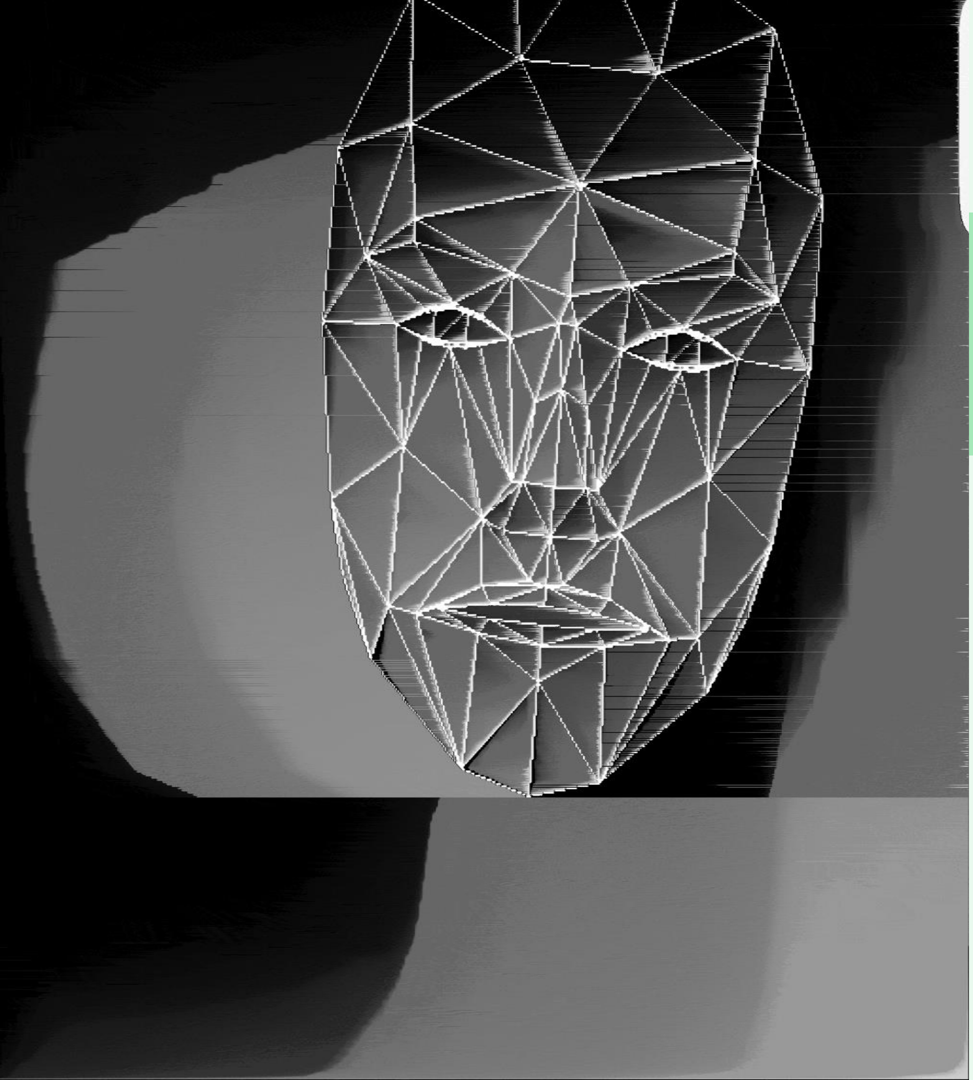


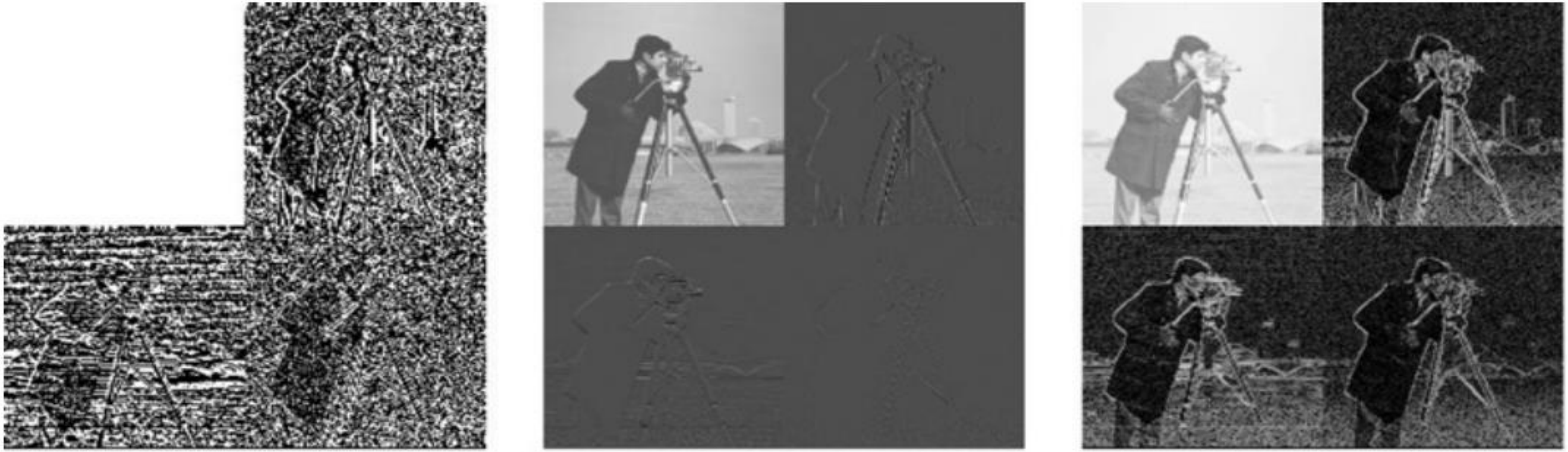
Fig: The standard decomposition of the Two-Dimensional DWT



04.

Wavelets and Images

Wavelets and Images



(a) No adjustment

(b) With `mat2gray`

(c) With log and `mat2gray`

Fig: Different displays of a 1scale DWT applied to an image

Wavelets and Images (Cont.)

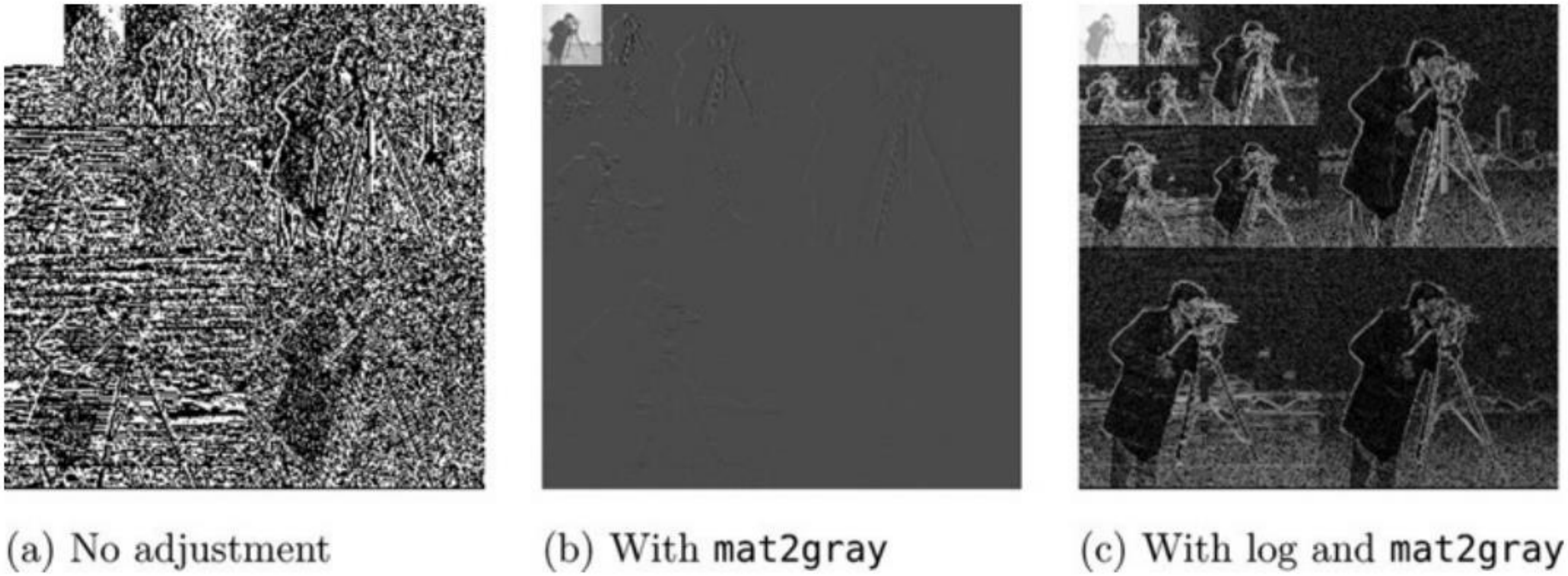
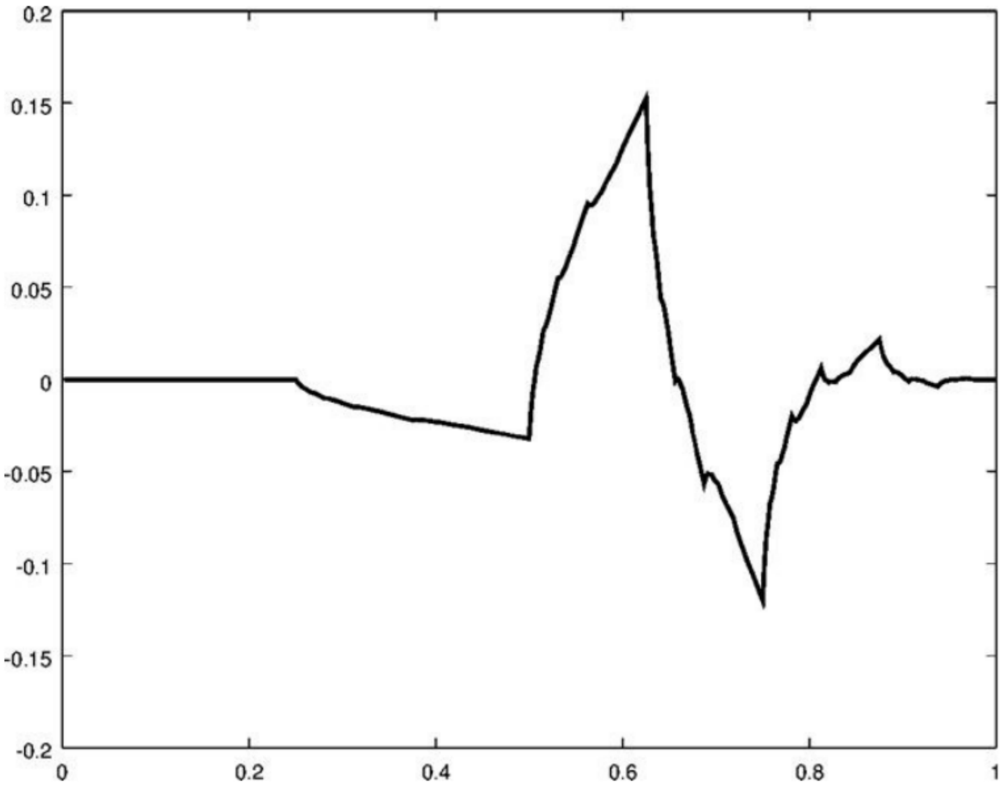


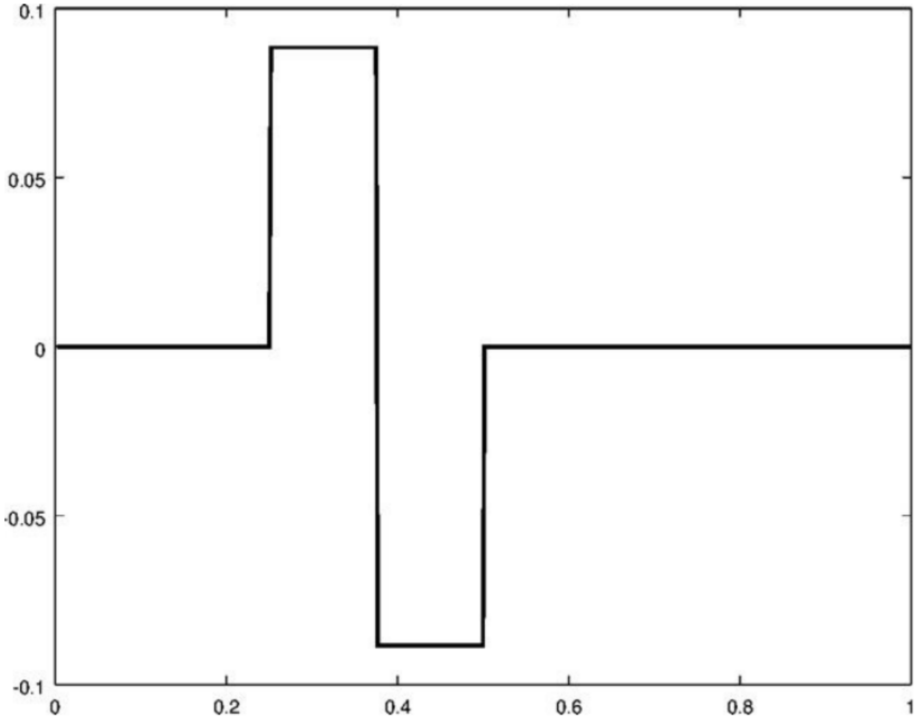
Fig: Different displays of a 3scale DWT applied to an image

A plot of the Daubechies4 wavelet



The Daubechies Wavelets

A plot of the Haar wavelet



The Haar Wavelet

05.

Image Compression

Image Compression using Wavelets

- Image compression using wavelets in digital image processing involves breaking down an image into different frequency components using wavelet transforms.
- These components represent various levels of detail. The process includes **quantizing the coefficients, reducing data precision, and encoding them using entropy coding**.
- This results in a significantly compressed data representation of the image. During decompression, the data is decoded, and the original image is reconstructed by reverse wavelet transformations.
- This technique balances compression ratios and image quality, making it suitable for efficient storage, transmission, and processing of images.

Thresholding and Quantization

The idea is to take the DWT of an image, and for a given value d, set all values x in the DWT for which $|x| \leq d$ to zero.

Fig: An image after DWT thresholding and inversion

DWT Extraction

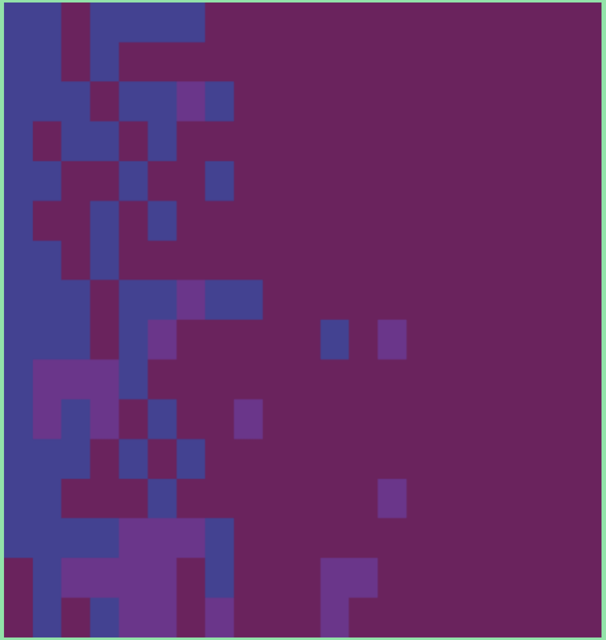
Here we cut off a portion of the DWT, setting all values to zero, and invert the rest. This is not a standard method of compression, but it does gives reasonably good results. It also has the advantage of being computationally easier than thresholding and quantization.

An image after DWT extraction and inversion

Compression by DWT extraction

06.
HIGH PASS FILTERING

High pass Filtering using Wavelets



High Pass Filtering using Wavelets

- High-pass filtering using wavelets involves applying a mathematical technique to emphasize or extract high-frequency details from a signal or image.
- It works by transforming the data using wavelets, isolating the high-frequency components, and then reconstructing the modified data.
- This process is useful for tasks like edge detection, texture analysis, and image enhancement, where highlighting fine details is important.

High Pass Filtering using Wavelets (Cont.)

```
# Generate a simple signal with low and high-frequency components
t = np.linspace(0, 1, 1000)
signal = np.sin(2 * np.pi * 10 * t) + 0.5 * np.sin(2 * np.pi * 100 * t)

# Perform the wavelet transform
wavelet = 'db4'  # Choose the wavelet (Daubechies-4 in this case)
coeffs = pywt.wavedec(signal, wavelet)

# Set the desired level of high-pass filtering
level = 2

# Modify the wavelet coefficients to perform high-pass filtering
for i in range(1, level + 1):
    coeffs[i] = np.zeros_like(coeffs[i])  # Remove low-frequency components

# Reconstruct the filtered signal using the inverse wavelet transform
filtered_signal = pywt.waverec(coeffs, wavelet)
```

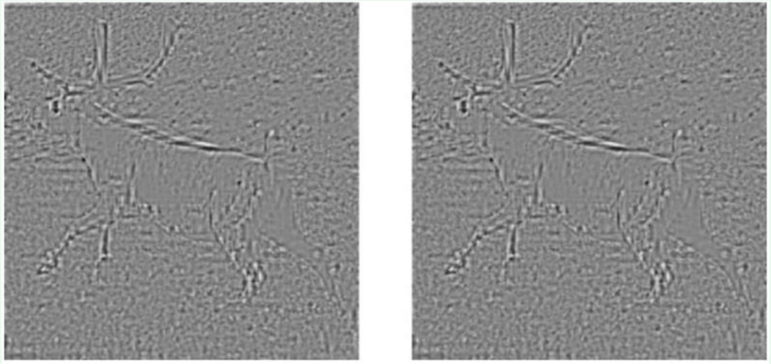


Fig: Output Obtained for High Pass Filtering using Wavelets

Python Code for High Pass Filtering using Wavelets

07.
Denoising

Denoising using Wavelets

Denoising

Denoising using wavelets is a technique in which wavelet transforms are employed to remove noise from signals or images. It involves the following steps:

- **Wavelet Transform**: Transform the noisy signal or image into a wavelet domain, decomposing it into different frequency components.
- **Thresholding**: Set a threshold to identify and remove coefficients associated with noise. Coefficients below the threshold are considered noise and are set to zero.
- **Inverse Wavelet Transform**: Reconstruct the denoised signal or image using the modified wavelet coefficients.

Denoising with wavelets effectively preserves important signal features while reducing unwanted noise. The choice of wavelet, thresholding method, and threshold level influence the denoising results.

Denoising (Cont.)

- For attempting some noise removal, we assumed that all the filter coefficients are those of the Daubechies-4 wavelet.

- The noisy image and the result after wavelet filtering are shown in Figure(A). At this level of thresholding, not much of the noise has been removed, although there has been some slight reduction.

- Thresholding can of course be done at higher levels:
  Figure(B) shows the results using (a) a threshold of 0.3 and (b) a threshold of 0.5. The results compare very favorably. However, we can see that too high a threshold results in unacceptable blurring of the image.

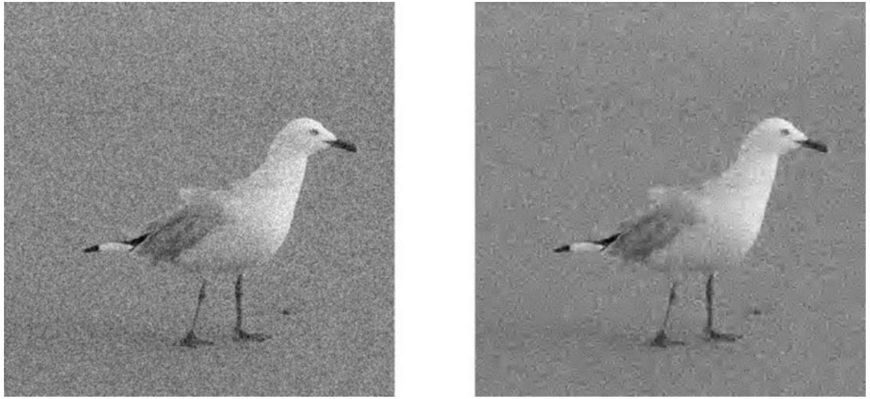
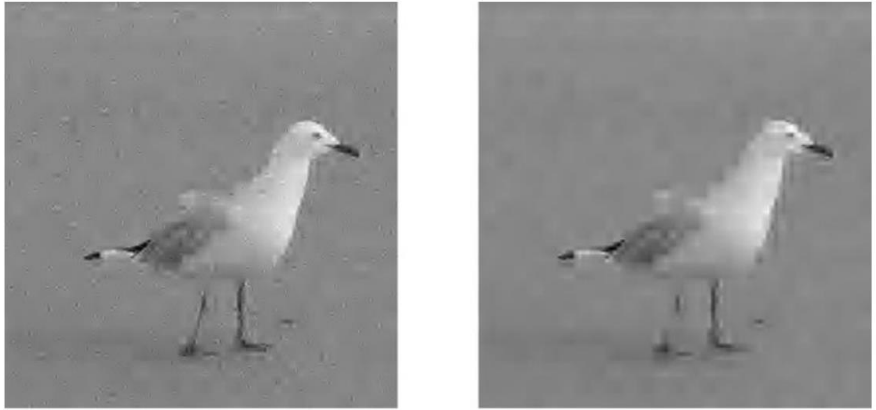
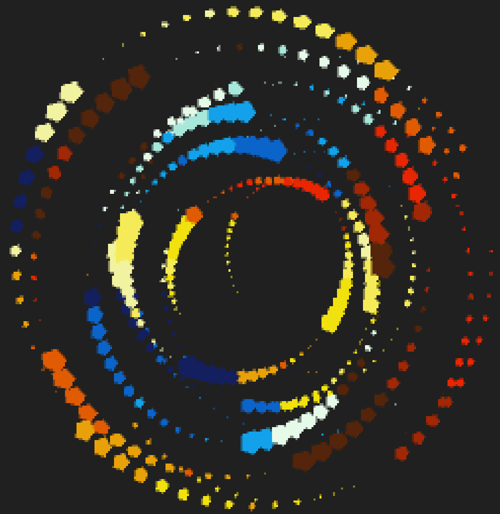


Fig (A): Denoising using Wavelets



Fig(B): Denoising using Different Thresholds

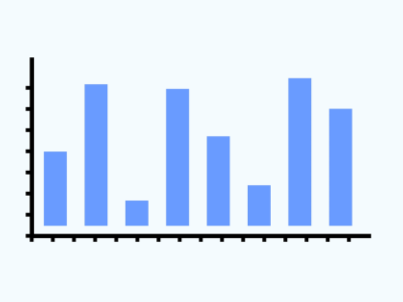


08. Conclusion

A summary to all that we have discussed

Conclusion (Cont.)

- In conclusion,
  Wavelets have emerged as a powerful and versatile tool in the field of signal and image processing.



- Throughout our discussion, we have explored the fundamental concepts, properties, and applications of wavelets, shedding light on their transformative impact on various industries and domains.

Thanks!

Questions? Comments? Let us Know