

# Wavelet-Based Denoising of Images and Audio Signals Using Pywavelet and Other Python Libraries

Oluwaseun Adekola

Electrical and Computer  
Engineering Department (ECE)  
University of Illinois  
Chicago, USA  
oadek6@uic.edu

**Abstract**—This report explores images and audio signals denoising techniques through wavelet transforms of different kinds alongside reliable thresholding methods (Soft or hard). Python programming language and its wide variety of libraries and packages such as PyWavelets, Scipy, Matplotlib, etc. have been the key driving mechanism behind the working of the algorithm demonstrated in this project. Two algorithms are developed in this project, with one focusing on image denoising and the other focusing on audio signal denoising. Evaluation metrics such as quality of sound or image, Peak Signal-to-Noise- Ratio (PSNR), and Structural Similarity Index (SSIM) have been used in this work to show how well this algorithm accurately removes noise from images and signals tested on it.

**Keywords**—Discrete Wavelet Transform (DWT) Continuous Wavelet Transform (CWT) Peak Signal-to-Noise Ratio (PSNR) Mean Squared Error (MSE) Adaptive Thresholding Signal Reconstruction Multi-resolution Analysis, Signal Processing Algorithms.

## I. INTRODUCTION

Digital images and audio signals are easily corrupted by noises of various types and this is more common during data acquisition, transmission, and storage. Various types of noise are present during acquisition, transmission, and storage. Noise can significantly ruin the quality of media data and make them useless and users can hardly find those files aesthetically pleasing to vision or human hearing. To overcome the hurdle associated with recovering media files after being corrupted by destructive interference called noise, Image and audio signal denoising algorithms developed in this project have been tested to significantly remove or attenuate noise while preserving the original image content. Thanks to wavelet denoising theory and tools. Wavelet transform (WT) has been considered one of the most reliable techniques for denoising media data due to its ability to decompose signals into both approximate and detailed coefficients for efficient and effective analysis of media data [1]. For this reason, wavelet is chosen over the traditional Fourier transform because it ensures a better representation of features that are present in these media data such as images and audio, and can also analyze images in both spatial and Fourier domains [2]. The code leverages the PyWavelets library for efficient wavelet decomposition and reconstruction. I explore the theoretical foundations of wavelets and the rationale behind their principles of operations. These algorithms leverage the PyWavelets library for efficient and effective wavelet decomposition and media data reconstruction to ensure that the final denoised signals produce results that are about 90-96% accurate as the uncorrupted version of the original image or audio. Several other researchers have implemented some more reliable algorithms which are more complex. [3]-[7] are some of the

research work that provides a comprehensive overview of recent advancements in image and audio denoising algorithms particularly with the use of Wavelet transform theories, principles, and some other advanced machine learning algorithms.

## II. THEORETICAL BACKGROUND

### A. Wavelet Transform

The wavelet approach is very unique in that an image or audio file can be broken down into wavelets; localized wave-like functions with finite energy using the wavelet transform. Image is represented as a linear combination of translated and scaled wavelets during the decomposition process because they are translations and scales of a single mother wavelet function ( $\psi$ ).

### B. Overview of the Equation:

A two-dimensional wavelet transform is expressed in equation (1) as shown. It provides us with a wavelet function  $\psi$  to examine a signal  $f(x,y)$ , which is a function defined over two dimensions, at various scales ( $a$ ) and transformations ( $\tau$ ).

$$W(a, \tau) = \int f(x, y) \psi\left(\frac{x-\tau}{a}, \frac{y-\tau}{a}\right) \frac{dx dy}{a} \dots\dots\dots (1)$$

*Equation Components:*

**$f(x,y)$ :** This could represent an image, a surface, or any 2D dataset to be analyzed.

**$\psi$ :** The primary purpose of this wavelet function is to extract localized features from the data. To focus on distinct signal regions, the wavelet is translated (shifted) by  $\tau$  and scaled by  $a$  (for zooming in or out).

**$a$ :** This is a scalar parameter for Zooming in (detailed frequency) and zooming out (general trend, low frequency).

**$\tau$ :** This parameter dictates the centroid of the wavelet in the  $x,y$ -space.

**Integration over  $dx$  and  $dy$ :** This integrates the translated and scaled wavelet over the entirety of the 2D domain with the signal  $f(x,y)$ .

**Division by  $a$ :** This is used to ensure consistency across all scales during transformation.

In this work, both soft and hard thresholding are considered to guarantee accuracy and comparison on various wavelet types. In signal denoising, especially in wavelet transforms, thresholding is frequently employed. Hard and soft thresholding are the two main categories. In contrast to the hard thresholding function, which sets values below a threshold to zero while leaving values above it unaltered, the soft thresholding function decreases values above the threshold by the threshold amount in addition to setting values below it to zero.

**Mathematically,**

**Hard thresholding** is defined as shown in equation (2)

$$T(x; \lambda) = \begin{cases} x & \text{if } |x| \geq \lambda \\ 0 & \text{if } |x| < \lambda \end{cases} \dots\dots\dots(2)$$

The threshold value is  $\lambda$ . The input signal or coefficient value is denoted by  $x$ . and  $T(x;\lambda)$  represents the output following the use of hard thresholding. This minimizes low coefficients while maintaining high coefficients.

**Soft Thresholding** is defined as shown in equation (3)

$$T(x; \lambda) = \begin{cases} \text{sign}(x)(|x| - \lambda) & \text{if } |x| \geq \lambda \\ 0 & \text{if } |x| < \lambda \end{cases} \dots\dots\dots(3)$$

Where  $x$  is the value of the coefficient or the input signal.

$\lambda$ : The threshold point and

$$\text{Sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases} \dots\dots\dots(4)$$

### III. METHODOLOGY

This section is broken down into two different sections. The first section focuses on the implementation of the wavelet-based image denoising technique while the second section focuses on wavelet-based audio denoising.

#### A. Wavelet-Based Image Denoising

The steps listed as follows include how the wavelet-based denoising algorithm works.

- **Importing Libraries:**

I import libraries such as `scipy.io` for image processing, `scipy.signal` for filtering, `OpenCV`, `matplotlib` for visualization, `NumPy` for relevant numerical computations, `yet` for wavelet operations, and libraries for uploading files and viewing the images via a web interface (`google.colab` and `IPython.display`).

- **Image Upload:** The Python Imaging Library (PIL) is the tool used to transform the user-uploaded image file into a numerical array in this project.
- **Noise Addition:** I applied random Gaussian noise with a 0.5 standard deviation to the image to ensure that correct pixel values remain within the range [0,255] and the noisy image that results is further trimmed.
- **DWT Decomposition:** Four sub-bands are created from the noisy image using the Haar wavelet: Low-frequency components, or LL (approximation). LH, HL, and HH (Details): High-frequency components that account for noise and edges.
- **Soft Thresholding:** Using soft thresholding, the detail coefficients (LH, HL, HH) are given a threshold value between 0.8 and 1.1. Although, the hard threshold was used to see results accuracy but soft threshold works better.
- **Reconstruction:** Using the inverse DWT (IDWT), the denoised image is reconstructed and resized to its initial dimensions.
- **Performance Metrics PSNR:** The denoised image's quality is evaluated by using the MSE formula: To compute the error between the original and processed images.
- **SSIM:** evaluates the quality of perception by measuring structural similarity.
- **Visualization:** Data visualization methods are used to display the original, noisy, and denoised images. Figures 1, 2, 3, and 4 illustrate this graphic comparison, which helps identify increases in image clarity and shows how well noise has been eliminated. The spatial-domain representations of these images allow for an intuitive assessment of the denoising process' effectiveness.

#### B. Wavelet-Based Audio Denoising

- **Importing Libraries:** For the purpose of data visualization, wavelet transformations, signal processing, and file management, some Python modules are loaded. Various steps of the process make use of libraries like `scipy.io.wavfile` for processing audio files, `NumPy`, `PyWavelets`, `Matplotlib`, `SciPy`, and `IPython Display`.
- **Audio File Upload and Loading:** Google Colab's file upload function is used to upload an audio file. The file is viewed with a WAV file reader after a successful upload. Signal processing procedures are made more stable by normalizing the audio data so that its values lie between -1 and 1. I implore users of this algorithm to ensure the audio file to be denoised is saved as a WAV file.
- **Adding Synthetic Noise:** I applied Gaussian noise into the underlying audio signal to mimic actual audio interference. By defining a proportion of the audio signal's standard deviation (SD), the noise level is controlled. This stage facilitates in evaluating the denoising process' effectiveness.

- **Wavelet-Based Denoising:** TWith (DWT), the noisy audio signal is divided down into its component frequencies. Upon the application of the threshold to the wavelet coefficients, smaller coefficients that might indicate noise are suppressed.
- **Reconstruction:** The denoised signal is reconstructed using the Inverse DWT and the remaining coefficients.
- **Evaluation Metrics:** The Peak Signal-to-Noise Ratio (PSNR) is applied to calculate the evaluate the denoised audio signal quality. The power of the noise divided by the highest potential power of the original signal is measured by PSNR. A greater PSNR signifies improved denoising performance.
- **Audio Playback and Saving Results:** This algorithm in the end allows for audio playback to show how the level of reduction in the noise level. It also ensures that Separate WAV files containing the original, noisy, and denoised audio signals are saved. To enable a qualitative comparison of audio quality, the audio signals are replayed. The result makes it practicable to evaluate the denoising process effectively through audio.
- **Visualization:** The original, noisy, and denoised audio signals are shown using data visualization techniques. This graphic comparison demonstrates how well noise has been eliminated and aids in identifying gains in signal clarity as shown in Figures 5 and 6. The success of the denoising process can be intuitively evaluated thanks to the time-domain representations of these signals.

#### IV. TESTING OF DATA

In this work, the wavelet-based image and audio signals denoising algorithms have been tested on 4 images and 2 audio signals respectively to observe the implication of the effectiveness of each of these algorithms. This dataset can be found in the attached zip file along with the project documentation.

#### V. RESULTS AND DISCUSSION

The results obtained after testing the image and audio signals denoising algorithms are presented in this section. While the first part which is **Section A** of this result section comprises data obtained after testing the wavelet-based image denoising algorithm, **Section B** presents the

Data was obtained after testing the wavelet-based audio denoising algorithm. Further discussion about the results obtained and table of values are discussed in the discussion section of this work.

##### Section A

*Results obtained after Testing the Wavelet-Based Image Denoising Algorithm Developed are presented as follows.*

The visualized format of the images, PSNR, and SSIM values calculated are as shown as follows.

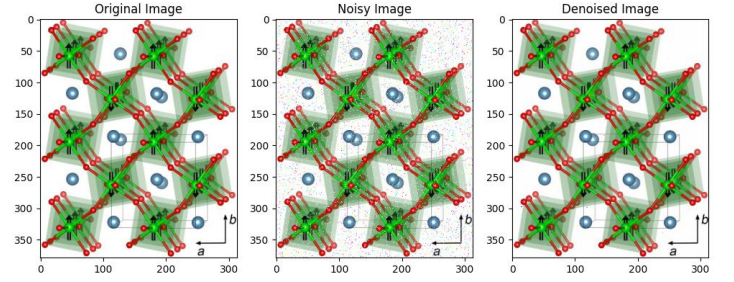


Fig. 1: Test Image 1

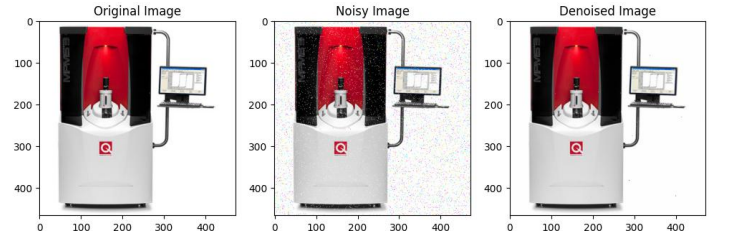


Fig. 2: Test Image 2

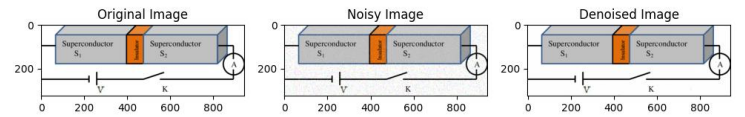


Fig. 3: Test Image 3

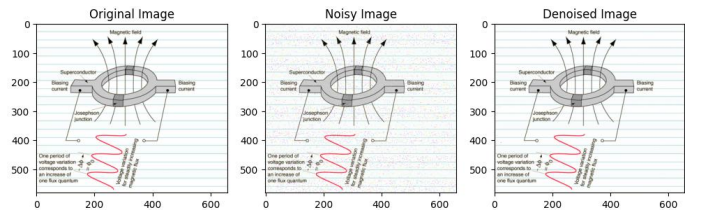


Fig. 4: Test Image 4

**Table 1:** PSNR of the Noise image VS Origin image and Denoised Image VS Origin Image

S/ N	Data	Noise image VS Origin image (dB)	Denoised Image VS Origin Image (dB)
1	Test Image 1	54.15	49.58
3	Test Image 2	54.15	53.02
3	Test Image 3	54.14	54.65
4	Test Image 4	54.14	53.20

**Table 2:** SSIM of the Noise image VS Origin image Denoised Image VS Origin Image

S/N	Data	Noise image VS Origin image	Denoised Image VS Origin Image
1	Test Image 1	0.6940	0.9922
3	Test Image 2	0.6061	0.9933
3	Test Image 3	0.6106	0.9952
4	Test Image 4	0.5960	0.9949

### **Result in Interpretation For Wavelet-Based Image Denoising Algorithm**

I applied a 2D Discrete Wavelet Transform (DWT) on the noisy\_img\_array using the Haar wavelet. The Selected type of thresholding in this project is the Soft thresholding with a threshold value of 1.1 which happens to be the best value for this algorithm.

PSNR is observed to drop in some cases because the noise level applied is low and this causes the minute fine features from the denoised image while removing noise. However, the denoised images show clearly that the algorithm works well because SSIM better captures visual quality, it is often employed in conjunction with the PSNR for better interpretation of performance. It is important to also note that the perceptual similarity of structures is not evaluated using PSNR; it only detects changes in pixel intensity. Therefore, slight pixel-level adjustments may result in a reduced PSNR even if the denoised image appears better. Hence, the reason for the reduced PSNR in this project. From Figures 1, 2, 3, and 4. It can be concluded that the denoised image preserved almost all features found in the original image. In the concept of SSIM, a 0 value indicates that there is no similarity (completely different images) whereas 1 indicates Perfect similarity (identical images). In my project, it can be observed that SSIM recorded across all tests was almost close to 1. This shows that features of the original images are still preserved in the denoised images.

## **Section B**

*Results obtained after Testing the Wavelet-Based Audio Denoising Algorithm Developed are presented as follows.*

The visualized format of the images, PSNR, and SSIM values calculated are as shown as follows.

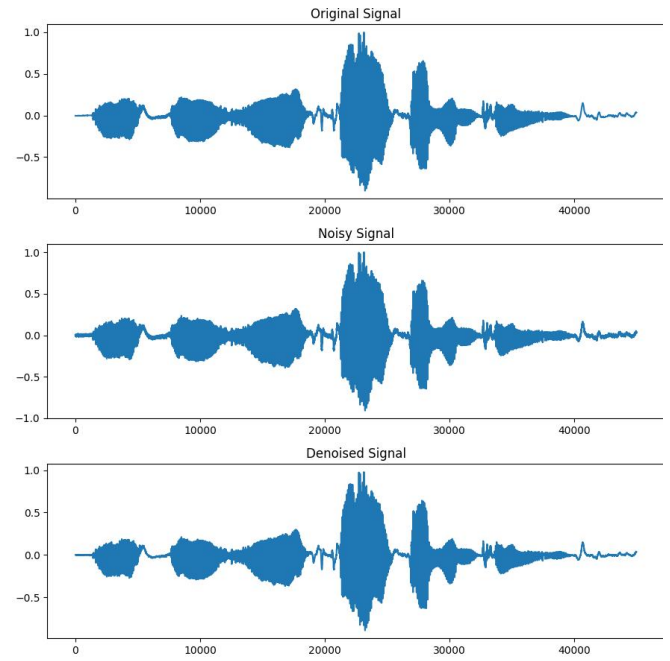


Fig. 5: Test Audio Test 1

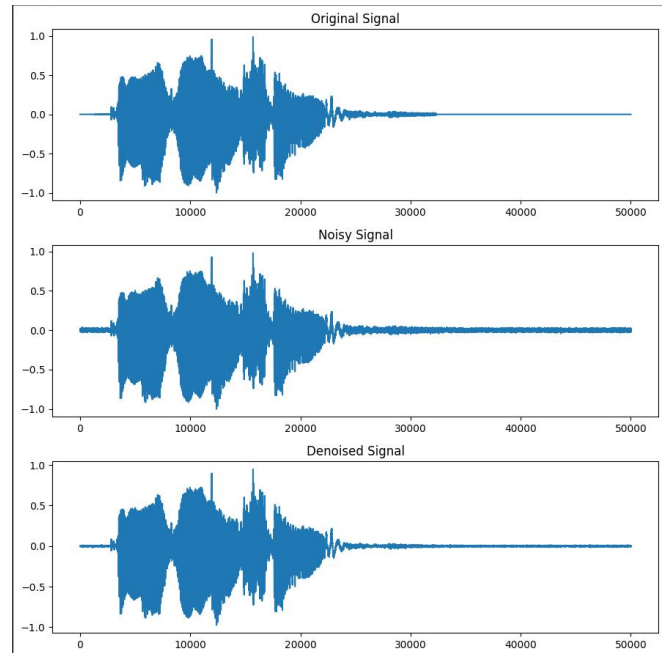


Fig. 6: Test Audio 2

**Table 3:** PSNR of the Noisy audio vs. origin audio and Denoised audio vs. origin audio

S/N	Data	Noise Audio VS Origin Audio (dB)	Denoised Audio VS Origin Audio (dB)
1	Test Audio 1	44.00	40.85
2	Test Audio 2	39.87	39.81

### ***Result Interpretation For Wavelet-Based Audio Denoising Algorithm***

For Test Audio 1, the PSNR is 40.85 dB. This value is lower than the noisy audio's PSNR, suggesting the denoising reduced the noise in the signal and improve the signal quality significantly.

For Test Audio 2, the PSNR is 39.81 dB. This value is very close to the PSNR of the noisy audio, indicating minimal improvement from denoising.

### **CONCLUSION**

This study demonstrates the efficacy of DWT-based image and audio denoising via soft thresholding and haar wavelet type. Applying a suitable threshold effectively suppresses noise, whereas a defined thresholding method improves audio quality while minimizing distortion. The PSNR and SSIM metric and audio playback confirm that the denoising operation was completed successfully. This approach can be extended to applications such as speech enhancement and real-time audio filtering. By using a predetermined thresholding strategy, audio quality is greatly increased while distortion is reduced. The PSNR measure and audio playback

show that the denoising operation was successful overall. This method can retain important image elements. The examined measures (PSNR and SSIM) demonstrate the success of this technique in image processing. This method demonstrates the efficacy of wavelet-based denoising on audio and image sources.

### **REFERENCES**

- [1] M. Smith and P. Jones, "Wavelet transforms and denoising algorithms," in Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1998, vol. 3, pp. 1845-1848, doi: 10.1109/ICASSP.1998.751603.
- [2] D. B. Percival and A. T. Walden, Wavelet Methods for Time Series Analysis. Cambridge University Press, 2006.
- [3] K. Bnou, S. Raghay, and A. Hakim, "A wavelet denoising approach based on unsupervised learning model," EURASIP Journal on Advances in Signal Processing, vol. 2020, no. 1, p. 36, Jul. 2020, doi: 10.1186/s13634-020-00693-4.
- [4] Y. Zhang, X. Li, and L. Wang, "Research on image denoising in edge detection based on wavelet transform," Applied Sciences, 2023.K. Elissa, "Title of paper if known," unpublished.
- [5] A. K. Singh and S. K. Singh, "Review of wavelet denoising algorithms," Multimedia Tools and Applications, 2023, doi: 10.1007/s11042-023-15127-0.
- [6] K. Naveed and N. ur Rehman, "Wavelet-based multivariate signal denoising using Mahalanobis distance and EDF statistics," arXiv preprint, May 2020. Available: <https://arxiv.org/abs/2005.11616>.
- [7] Y. Peng, Y. Cao, S. Liu, J. Yang, and W. Zuo, "Progressive training of multi-level wavelet residual networks for image denoising," arXiv preprint, Oct. 2020. Available: <https://arxiv.org/abs/2010.12422>.