Wavelet Transform for Image Processing

Zhang Jinrui[1]
He Jiashun
Meng Jingyuan
Jiang Zishen
Mo Zian

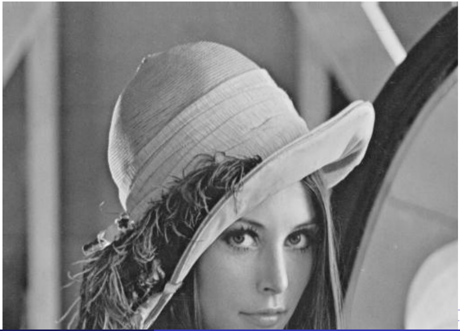
August 6, 2025

[1] alternative email:zhangjr1022@mails.jlu.edu.cn

1. Image Loading and Grayscale Conversion

Step Explanation

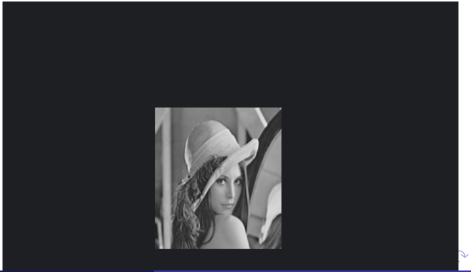
The image is loaded using a Python imaging library and converted to grayscale.
This reduces data complexity and focuses analysis on structural content.



2. Rescaling and 3. Centered Cropping

Rescaling with Aspect Ratio Preservation

The image is resized using high-fidelity interpolation to ensure one side reaches the target length (2^N), without distortion.
The resized image is cropped to $2^N \times 2^N$, ensuring input uniformity without compromising important visual features.



4. Matrix Output

Step Explanation

The final image is converted into a matrix format suitable for:

- Mathematical operations (e.g., wavelet transforms)
- Storage and machine learning integration

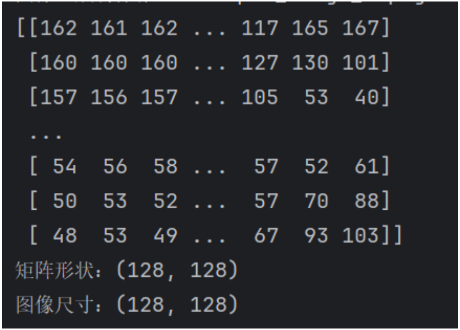
128	135	142
130	138	145
125	132	139

Example of 3x3 image matrix (simplified)

5. Convert Matrix Information into an Image

Concept

Each matrix element represents the gray value of a pixel. Using this matrix, we can reconstruct the grayscale image.



```
[[162 161 162 ... 117 165 167]
 [160 160 160 ... 127 130 101]
 [157 156 157 ... 105  53  40]
 ...
 [ 54  56  58 ...  57  52  61]
 [ 50  53  52 ...  57  70  88]
 [ 48  53  49 ...  67  93 103]]
矩阵形状：(128, 128)
图像尺寸：(128, 128)
```

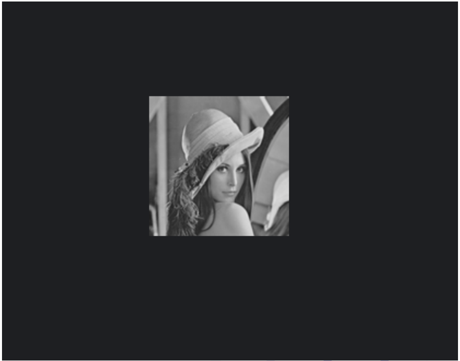


Figure: Matrix Information

Standard Haar Decomposition

- sdlfkj
- sDF
- sdf

sfdfg

- sdlfkj
- sDF
- sdf

sadsgkjh

- sdlfkj
- sDF
- sdf

sdf

- sdlfkj
- sDF
- sdf

Basic Idea

- $$\sum_{k=1}^{n} \hat{l}_k^4 \geq (\sum_{k=1}^{n} \hat{l}_k^2)^2 = (\sum_{k=1}^{n} l_k^2)^2$$

- $$\min_{\hat{l}_k \in l^2 st. \sum_{k=1}^{n} \hat{l}_k^2 = \sum_{k=1}^{n} l_k^2} (-\sum_{k=1}^{n} \hat{l}_k^4)$$

Basic Idea

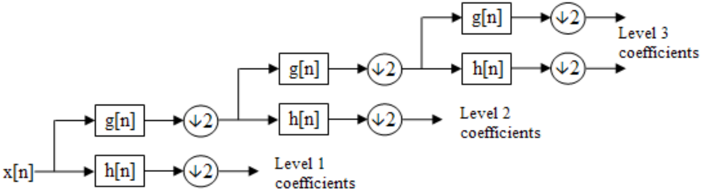
- 

Figure: Cascading

- $$\min_{\Phi}(-\sum_{k=1}^{n}\hat{l}_k^4)$$
- Condense the energy to as less coefficients as possible.

Results

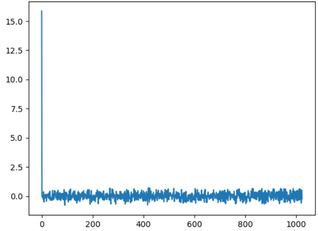


Figure: Haar 2x2 filter bank random input. Compression Rate 0.2568

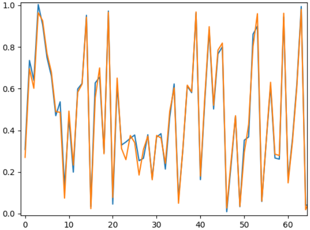


Figure: Haar 2x2 filter bank random input.Total average energy loss 0.0009

- Random sequence would have large information entropy so it have a low compression rate.

Results

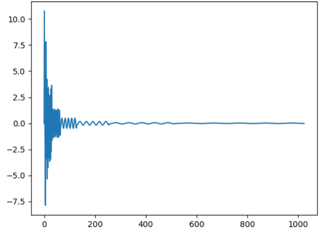
- 

  Figure: Haar 2x2 filter bank sin input frequency. Compression Rate 0.9287

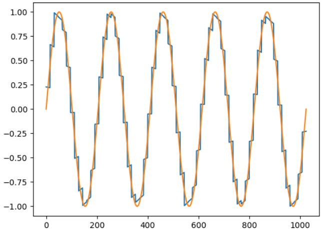
  

  Figure: Haar 2x2 filter bank sin input reconstruction.Total average energy loss 0.0104

- A smooth signal have small information entropy, so will have a high compression rate.

Results

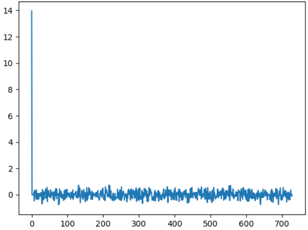
- 

Figure: Haar 3x3 filter bank random input. Compression Rate 0.1906

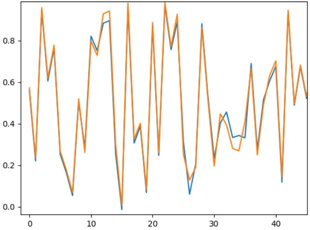


Figure: Haar 3x3 filter bank random input.Total average energy loss 0.0008

- same random sequence same reason.

Results

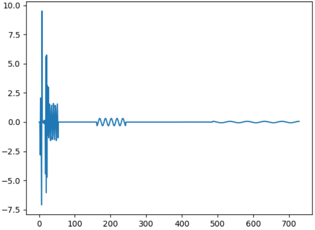
- 

Figure: Haar 3x3 filter bank sin input frequency. Compression Rate 0.7750

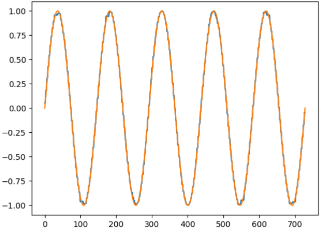


Figure: Haar 3x3 filter bank sin input reconstruction.Total average energy loss 0.0007

- same smooth signal same reason.

sadsgkjh

- sdlfkj
- sDF
- sdf

sdf

- sdlfkj
- sDF
- sdf