Image Compression

Goals and Structure

Please go through the following files in order.

1. *basic_wavelet_compression.m*: implement a basic wavelet compression and plot different levels.
2. *wavelet_packet_compression.m*: use wavelet packet to compress images and compare the results with the previous basic one.
3. *different_entropy_and_threshold_wavelet_packet.m*: run this file to explore how different threshold and entropy types affects the wavelet packet compression performance.

Theory

Wavelet Image Compression

Wavelet image decompose the image into a set of coefficients representing different frequency components.

1. **Wavelet Transform**:
   - An image is decomposed into a series of subbands using a wavelet transform (e.g., Discrete Wavelet Transform, DWT).
   - Each subband represents a different frequency component of the image, capturing details at various resolutions.
2. **Thresholding**:
   - Coefficients below a certain threshold are set to zero, reducing the amount of data to store.
   - Thresholding can be hard (setting coefficients below the threshold to zero) or soft (reducing the magnitude of coefficients towards zero).
3. **Quantization**:
   - Remaining coefficients are quantized to reduce the number of bits required for storage.
4. **Encoding**:
   - Quantized coefficients are encoded using techniques such as Huffman coding or run-length encoding to further compress the data.
5. **Reconstruction**:

   ```
   - The compressed image can be reconstructed by applying the inverse wavelet transform to the thresholded coefficients.
   ```

   We focus on 1, 2 and 5 in this project. Adding in 3 and 4 (Quantization and Encoding) will further increase the compression capability.

Wavelet Packet Image Compression

Wavelet packet image compression extends wavelet compression by allowing decomposition of both approximation and detail coefficients.

1. **Wavelet Packet Transform**:
   - Similar to wavelet transform, but both approximation and detail subbands are further decomposed, allowing for a richer analysis of the image frequencies.
2. **Thresholding**:
   - Coefficients are thresholded to remove less significant information.
3. **Entropy-Based Compression**:
   - Various entropy measures (e.g., Shannon entropy, log energy entropy) can be used to adaptively select the best basis for representing the image, optimizing the trade-off between compression and image quality.
4. **Quantization and Encoding**:
   - Quantization and encoding are applied to the thresholded coefficients, similar to basic wavelet compression.
5. **Reconstruction**:

   ```
   - The compressed image is reconstructed by applying the inverse wavelet packet transform.
   ```

We focus on 1, 2, 3 and 5 in this project. Adding in 4 (Quantization and Encoding) will further increase the compression capability.

Results

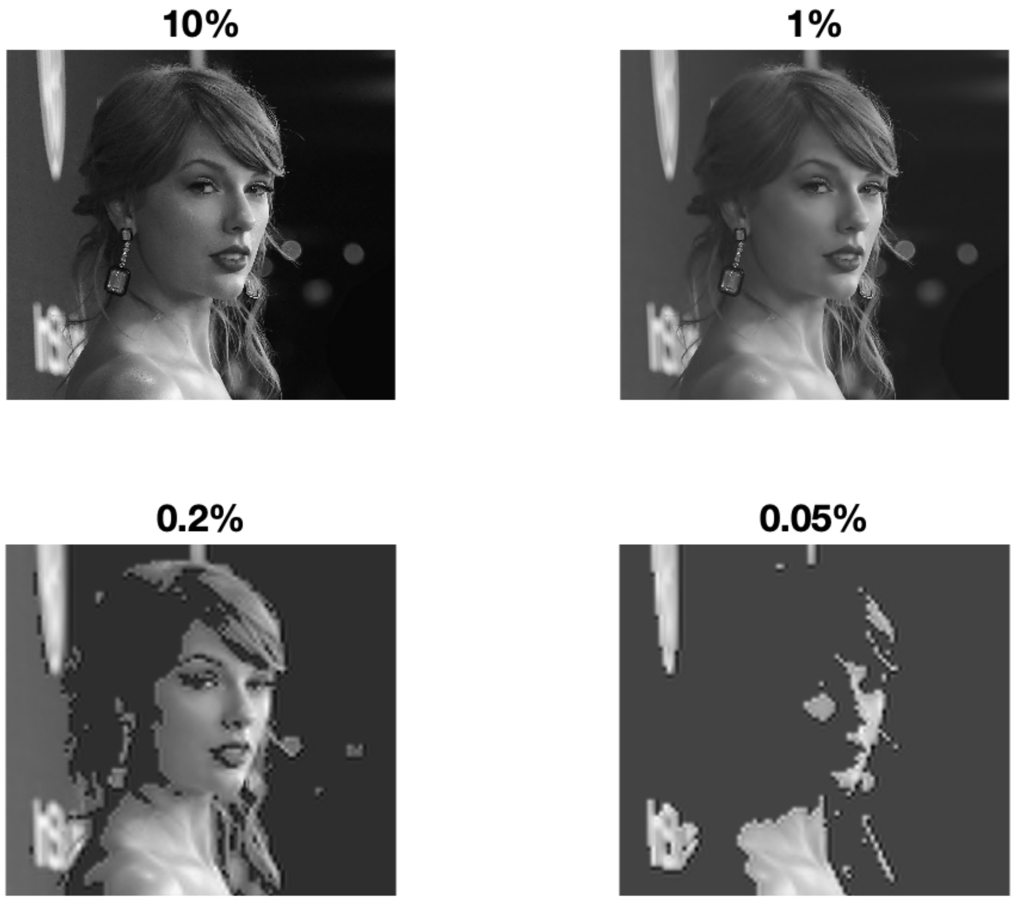
I picked some results to analyze here, to see the full results, you can either:

- run the programs, since I've added menus, you can play around with different parameters.
- see the *"Results"* folder in my submission.

basic_wavelet_compression.m:
Let's start with the classical example, using *db4* at *level* 4, wavelet compression.
here, compression score = the percentage of thresholded coefficients that are equal to 0, is a metric of *how much space can your algorithm save*.
and remaining energy = percentage of energe preserved, is a metric of *how well is the image preserved.*

I also tested different thresholds:
here we use *db4* at *level* 4 with a hard threshold,

x% means: top x percents of the wavelet coefficients (sorted by magnititude) remain unchanged, other small coefs are set to 0.



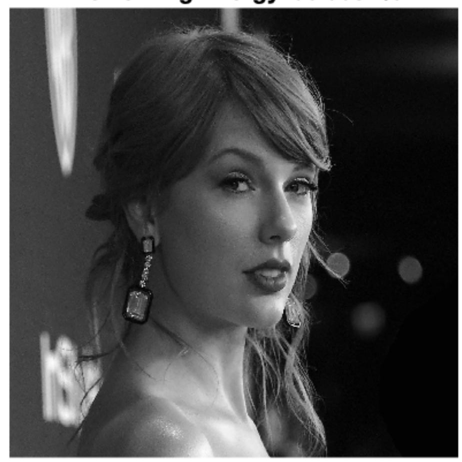
wavelet_packet_compression.m:
we compare *wavelet packet compression* with *wavelet compression* here.
observations:

- *wavelet packet compression* has a higher compression score.

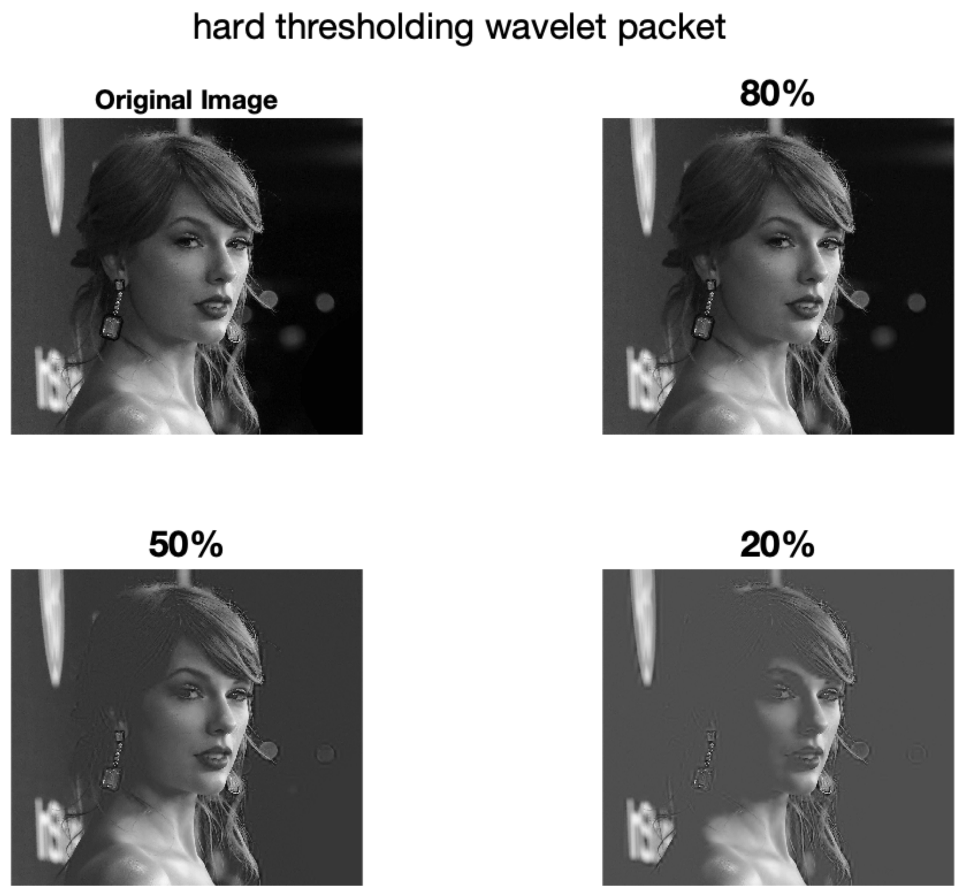
- they have roughly the same remaining energy.

Wavelet Packet at level 2 using db8
Compression score: 67.05%
Remaining Energy: 99.9934%

Wavelet at level 2 using db8
Compression score: 49.35%
Remaining Energy: 99.9984%



the following is more interesting.



wavelet packet compression performs poorly with a hard threshold and default Shannon entropy!

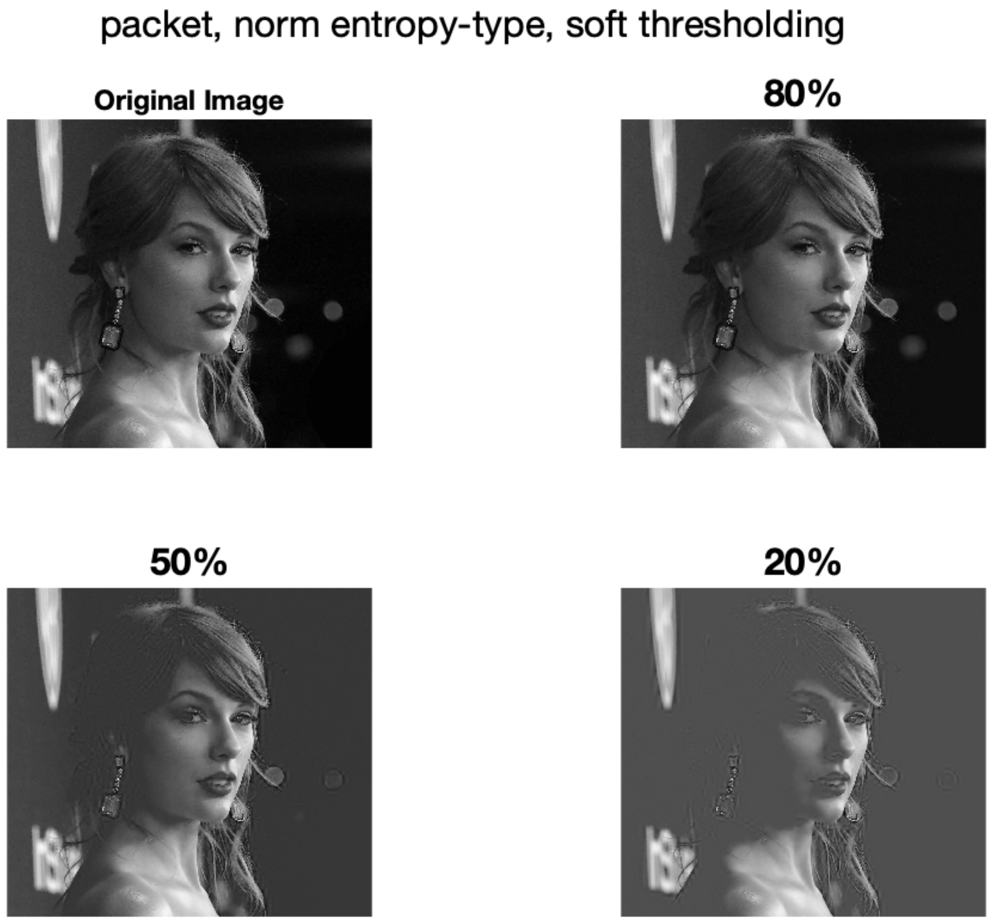
But why?

- *Shannon entropy encourages a balanced coefs sequence, whereas in basic wavelet compression, most of the info are concentrated in a small subset of coefs.*
- *A hard threshold kills too many coefs*.

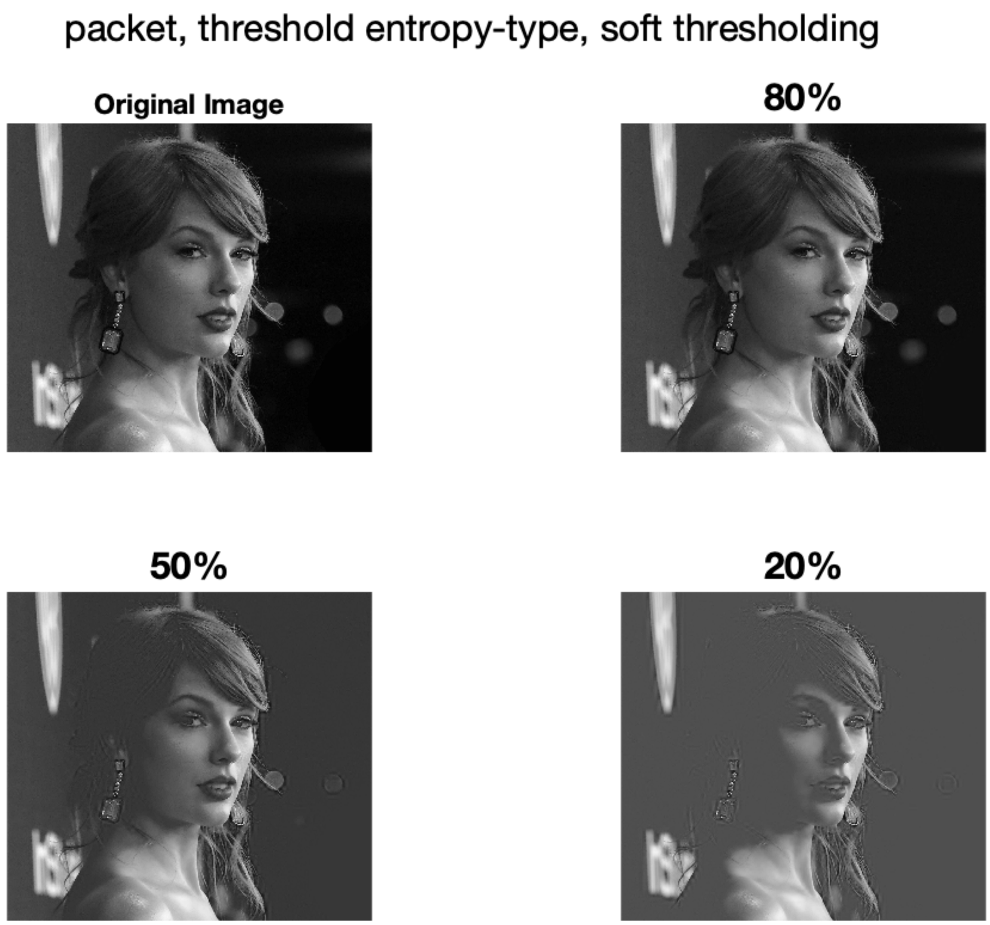
So, the idea is:

a soft threshold and a different entropy that promotes sparsity (such as the L1 norm entropy), will have better performance. This is exactly what the next file is about.

different_entropy_and_threshold_wavelet_packet:



packet, shannon entropy-type, soft thresholding



They are definitely better, but still not as good as the basic wavelet one...