Vector Quantization for Image Compression

DSAI 325 – Introduction to Information Theory
Assignment 4 Report
Student Name: Mohamed osama mahmoud darwish
Student ID: 202201273

---

1. Introduction

This assignment aimed to implement vector quantization using Java for image compression, following the Linde–Buzo–Gray (LBG) algorithm with codebook splitting and refinement. The main goal was to understand how vector/block size and codebook size affect the quality and efficiency of image compression.

Compression performance was evaluated based on **compression ratio** and **Mean Square Error (MSE)**. Several enhancements were implemented, including an optimized method for initializing the codebook, which improved convergence and reconstruction quality.

---

2. Methodology

Preprocessing

- Input: RGB image converted to grayscale.
- Image divided into fixed-size, non-overlapping blocks (default: 4×4).

Compression Process

- Blocks grouped using the LBG algorithm.

- A codebook of K vectors was built through splitting and refinement.
- Each image block was assigned to its nearest codeword (based on Euclidean distance).
- The compressed representation was stored using codeword indices and the codebook.

Decompression Process

- Original grayscale image was reconstructed by replacing code indices with corresponding vectors from the codebook.

Metrics Used

- **Compression Ratio:** Ratio of original size to compressed size (higher is better).
- **MSE (Mean Square Error):** Measures the distortion between original and decompressed images (lower is better).

---

3. Results and Analysis

Below is a summary of the results collected from multiple runs using different configurations:

Test	Original Size (bits)	Compressed Size (bits)	Compression Ratio	MSE	Notes
T1	9,830,400	388,096	25.33	255.62	Large image, strong compression
T2	524,288	24,576	21.33	103.89	Optimized codebook
T3	524,288	24,576	21.33	217.04	Optimized codebook
T4	524,288	24,576	21.33	292.02	Optimized codebook
T5	524,288	24,576	21.33	1081.14	Attempted adaptive block size

---

4. Discussion

- The highest compression ratio (25.33) was achieved with the largest image, demonstrating how larger images benefit more from block quantization.

- A consistent compression ratio of 21.33 was observed across most small-image tests.

- MSE values varied depending on the complexity of the image and codebook accuracy. The best visual reconstruction was achieved when MSE was below 200.

- The test using **adaptive block sizes** resulted in a significantly higher MSE (1081.14), indicating that naive block size variation can severely degrade

quality unless intelligently managed.

---

5. Optimizations Applied

To improve the performance and accuracy of the compression process, the following enhancements were made:

- **Smart Codebook Initialization**: Instead of naive ±1 splitting, the codebook was initialized using a distance-maximizing strategy inspired by K-Means++ to ensure better spread among initial centroids.

- **Experimental Adaptive Block Sizes**: Basic logic was added to experiment with different block sizes based on image characteristics. However, this approach caused high reconstruction error and requires further refinement.

These improvements contributed to better compression quality in tests T2–T4, with lower MSE and similar ratios compared to the base implementation.

---

6. Conclusion

Vector quantization proved to be an effective method for grayscale image compression. With proper block sizing and codebook refinement, high compression ratios can be achieved with acceptable loss in quality. Smarter codebook initialization noticeably improved clustering and MSE.

Further development, such as fully adaptive block sizes or deep-learning-assisted codebook prediction, could improve compression performance even more.

=== Compression Info ===
Original size: 9830400 bits
Compressed codes size: 384000 bits
Codebook size: 4096 bits
Total compressed size: 388096 bits
Compression ratio: 25.33

MSE: 255.61743408203125

=== Compression Info ===
Original size: 524288 bits
Compressed codes size: 20480 bits
Codebook size: 4096 bits
Total compressed size: 24576 bits
Compression ratio: 21.33
MSE: 103.89260864257812

=== Compression Info ===
Original size: 524288 bits
Compressed codes size: 20480 bits
Codebook size: 4096 bits
Total compressed size: 24576 bits
Compression ratio: 21.33
MSE: 217.0413818359375

=== Compression Info ===
Original size: 524288 bits
Compressed codes size: 20480 bits
Codebook size: 4096 bits
Total compressed size: 24576 bits
Compression ratio: 21.33
MSE: 292.0205383300781

=== Compression Info ===
Original size: 524288 bits
Compressed codes size: 20480 b
Codebook size: 4096 bits
Total compressed size: 24576 bits
Compression ratio: 21.33
MSE: 1081.1441955566406