Vector Quantization-Based Image Compression

Hamza Abdelmoreed *

*Zewail City of Science and Technology
Email: s-hamza.fekry@zewailcity.edu.eg

***Abstract*—This project focuses on image compression utilizing the technique of Vector Quantization (VQ). The goal is to achieve efficient image compression and analyze the performance using two different color spaces: RGB and YUV. The primary objectives include building an image compression model using VQ and evaluating it based on the Mean Squared Error (MSE) and Compression Ratio (CR) for RGB and YUV color spaces, respectively.**

***Index Terms*—Image Compression, Vector Quantization, RGB, YUV, MSE, Compression Ratio.**

I. Introduction

The goal of this project is to apply Vector Quantization (VQ) for compressing color images. The data set includes images from three categories: Nature, Faces, and Animals. The project focuses on two main objectives:

- Developing an image compression model using Vector Quantization (VQ).
- Comparing the compression performance between RGB and YUV color spaces based on Mean Squared Error (MSE) and Compression Ratio (CR).

II. Methodology

A. Codebook Generation

For both RGB and YUV color spaces, the following process was followed:

- RGB Color Space: Three separate codebooks are generated for the Red, Green, and Blue channels.
- YUV Color Space: The image is first converted into YUV format. The U and V channels are then subsampled (downscaled), while the Y channel remains unaffected.
- K-means Clustering: The codebooks are generated by applying K-means clustering on 2x2 blocks of image data, using the average of image blocks as the initial centroid.
- Codebook Size: Each codebook contains 256 vectors, each representing a 2x2 pixel block.

B. Image Compression

Once the codebook is generated, the image compression is done as follows:

- The image is split into 2x2 blocks.
- For each block, we find the closest matching vector in the corresponding codebook (RGB or YUV).
- The image is then compressed by storing the indices of the closest vectors from the codebook.

C. Image Reconstruction

For reconstructing the image:

- The compressed image data (indices of codebook vectors) is mapped back to the corresponding vectors in the codebook.
- The reconstructed image is generated by assembling these blocks back together.

III. Results

A. Image Compression Results (Sample)

RGB Compression Results:

- Compression Ratio (CR): 4.00
- Average MSE: 21.71

YUV Compression Results:

- Compression Ratio (CR): 8.00
- Average MSE: 44.03

These results suggest that YUV compression provides a higher compression ratio but leads to a higher MSE, indicating that the quality is slightly compromised for better compression.

B. Sample Image Results

Image	RGB - MSE	RGB - CR	YUV - MSE	YUV - CR
Image 0	15.75	4.00	26.00	8.00
Image 1	51.54	4.00	69.91	8.00
Image 2	46.63	4.00	61.65	8.00
Image 3	14.53	4.00	24.23	8.00
Average	21.71	4.00	44.03	8.00

TABLE I
Sample Results for RGB and YUV Compression

IV. Discussion

A. Compression Ratio

The YUV compression achieved a higher compression ratio (8.00), as expected, due to the subsampling of the U and V channels. This reduces the image size without a significant loss in overall information. However, this reduction comes with a trade-off in image quality.

B. MSE (Image Quality)

The RGB compression resulted in lower MSE, indicating better retention of image details. This is because no subsampling occurs in RGB, and each channel is compressed independently.

C. *Efficiency vs. Quality Trade-off*

The trade-off between compression and quality is apparent:

- YUV Compression: More efficient but lower quality.
- RGB Compression: Offers better quality but with a lower compression ratio.

V. Conclusion

In this project, successfully applied Vector Quantization for image compression. analyzed and compared the performance of the RGB and YUV color spaces. The results indicate that YUV compression is more efficient in terms of compression ratio but sacrifices some image quality. In contrast, RGB compression maintains higher quality at the cost of a lower compression ratio.

Future work could explore alternative compression methods like Huffman coding or deep learning-based compression for further optimization.