Evaluating the Effectiveness of The Representative K-Means Method for Vector Quantization in Image Compression

Krystal Ly

February 2023

Abstract

This project evaluates the effectiveness of the Representative K-Means Method of Vector Quantization in Image Compression. Image compression is a data compression technique used to reduce the size of digital image files without compromising the visual quality. One method of lossy image compression is vector quantization, which works by grouping similar pixels together and replacing them with a shared representation, called a "prototype vector". The compressed image is made up of the indices of these prototype vectors in a codebook. K-means Clustering algorithm can be used to generate prototype vectors that are representative of the original vectors in the image. The algorithm divides the vectors into several clusters based on their similarities, and each cluster has a center that can represent the characteristics of all vectors within that cluster. These centers can then be used as the prototype vectors. However, the K-means algorithm produces prototype vectors that are customized to a specific image, and they do not generalize well to unseen images. To avoid the inefficiency of applying K-means clustering to every image for compression purposes, we propose an alternative method known as the Representative K-Means Method for Vector Quantization (or simply, the Representative method). The method divides the set of images into different groups, chooses a representative image for each group, runs the K-means Clustering algorithm on those representative images, and applies those codebooks generated by the representative images to other images that are in the same group. We evaluate the effectiveness of this approach, on different sets of images, including the Carrot set, the Human set, and the Human Black and White set. Our results suggest that the effectiveness of the Representative method is inversely proportional to the variation in colors within each image and between images in a set of images. This project provides insights into when it is appropriate to use the Representative method and when it is not appropriate to do so.

1 Introduction

1.1 Image Compression

Image compression refers to a type of data compression applied to digital images. Its goal is to decrease the size of an image file without significantly compromising its visual quality to reduce the cost of storage or transmission. There are two main types of image compression: lossless compression and lossy compression. As its name suggests, lossless image compression is capable of reducing the size of an image while maintaining the original image quality. Lossless compression is useful for images that require exact reproduction, such as medical images, scans of important documents, pieces of evidence, and so forth. In real life, lossy image compression is more commonly used. It is a method of reducing the size of an image file by removing some of the less important information in the image. Lossy image compression can achieve extremely high compression rates based on how much information we are willing to sacrifice.

1.2 Vector Quantization

One method of lossy compression is vector quantization, in which the receivers and senders agree on a codebook of prototype vectors and encode the original vectors, or pixels, in an image to those prototype vectors. With vector quantization, instead of transmitting all data, the sender just needs to transmit the indices of the prototype vectors in the codebook, and the receiver can look up the indices in the codebook and reconstruct the image. Figure 1 illustrates vector quantization in more detail. Since the reconstructed image is made up of just the prototype vectors and not the exact vectors from the original image, the process sacrifices some of the information in the image (notice the difference between the color in the original and the reconstructed image in Figure 1). To preserve as much of the important information in the image as possible, it is necessary to choose prototype vectors that are representative of the original vectors that map to them. The K-means Clustering algorithm is one of the algorithms that can solve this problem of choosing representative vectors.

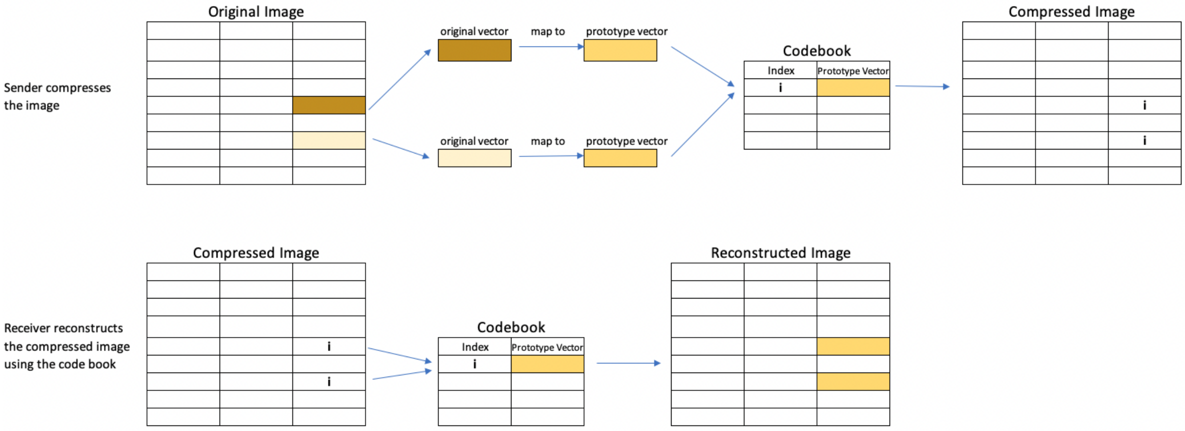

Figure 1: Illustration of Vector Quantization

1.3 Vector Quantization using K-means Clustering

K-means Clustering is a widely used method for implementing vector quantization in image compression. It partitions the vectors into a predetermined number of clusters based on their similarity. In Algorithm 1, we provide the pseudocode for the K-means Clustering Algorithm.

Algorithm 1: K-Means Clustering Algorithm

```
Select value for k, the number of centers
Initialize centers by selecting k random training points
Repeat
    #assignment phase
    for each training point s_i
        compute the distance of s_i to each center
    assign s_i to the closest center

    #adjustment phase
    for each center C
        compute the mean of all samples assigned to C
        move C to the location of the mean
Until no centers move
```

For each group of vectors, the center is the average point of the group, capable of representing some characteristics of all vectors in the group. The resulting cluster centers are used as the codebook entries to compress and reconstruct the image. Due to the nature of the K-means Clustering Algorithm, if the algorithm is run on a specific image, the centers, or codebook entries, are chosen in a way that is customized to that specific image. Therefore, when a codebook of a specific image is used on other images, it may result in an undesirable result or a greater loss of information than what is expected. The prototype vectors produced by the K-means Clustering Algorithm do not generalize well on unseen images but it is ineffective to run K-means Clustering on every image just to compress it. An alternative approach, says the Representative K-Means Method for Vector Quantization (or simply, the Representative method), is used to balance between time efficiency and accuracy.

1.4 Problem Statement

In this project, we propose an alternative method and evaluate its effectiveness in increasing time efficiency while maintaining the quality of the reconstructed image. Given a set of images, the method first divides the set of images into different groups and chooses a representative image for each group. It then runs the K-means algorithm on those representative images and applies the codebooks generated by the representative images to other images in the same group. This method is supposed to maintain the quality of the reconstructed images because images are compressed using the codebook generated by the representative image in its group. In addition, it overcomes the time inefficiency of having to run K-means on every image in the set. More details and illustrations of the method will be provided in the next section. This project conducts experiments to test the hypothesis that the effectiveness of the Representative method is inversely proportional to the variation in colors between images in a set of images, as well as the variation in colors within each image. Specifically, we hypothesize that if there is significant variation in colors between images and within each image, the application of the Representative method will lead to a significant decrease in image quality, and, therefore, will not be appropriate. To test this hypothesis, we used three sets of images - Carrot, Human Black & White, and Human - with varying degrees of color variation between images and within each image. We compressed the images in each set using the Representative method and evaluated the reconstructed images. The rest of the project is organized as follows. Section 2 provides a detailed explanation of the Representative method. It also introduces three sets of images and the implementation steps in more detail. In section 3, we report and analyze the results. The conclusion appears in section 4.

2 Methodology

2.1 The Representative Method

Figures 2 and 3 serve as illustrations of the Representative method. To begin with, the Representative method first selects representative images in a set of images using the K-means algorithm. In Figure 2, we use K-means to create two groups of images from the set of animal images. The images that are closest to the two centers are chosen as the representative images. The method then runs the K-means algorithm on each representative image to choose prototype vectors from those images. The prototype vectors are then assigned indices, becoming the codebooks (see Figure 3). The images in each group are compressed using the codebook generated by its representative image.

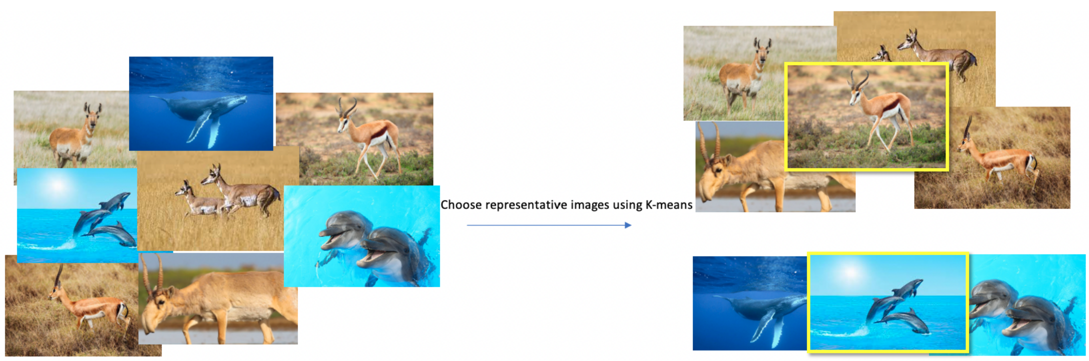

Figure 2: Illustration of the Representative method part 1

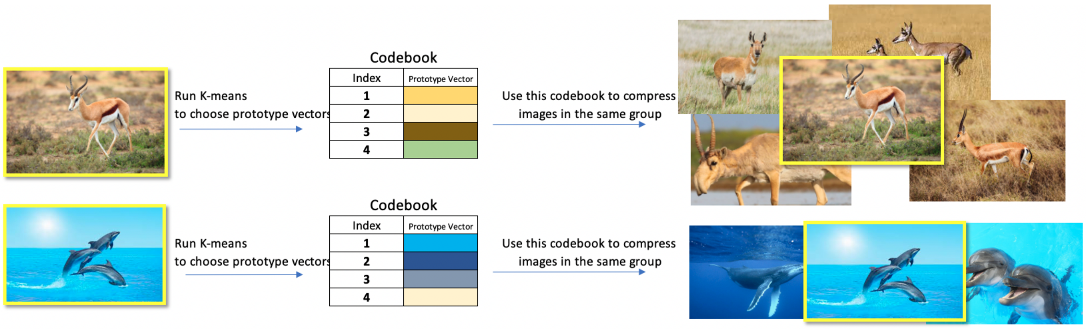

Figure 3: Illustration of the Representative method part 2

2.2 Datasets

The two sets of images that are used in this project are the Carrot set, a subset of the Vegetable Classification Dataset [1], and the Human set, a subset of the Face Mask Classification Dataset [2]. The Carrot set consists of images of carrots, and it is chosen because of the similarity in colors among all the images. The Human set consists of images of human portraits, and it is chosen because of the variation in colors among the images. Another set of images called the Human Black and White set is also used in this project. This dataset is generated by converting the original RGB-color images to grayscale images. To maintain a fair comparison between different sets of images, all sets of images consist of 200 images which are randomly chosen from the original datasets. Furthermore, all images will be resized to 224 x 224 pixels. Figures 6, 7, and 8 in the Appendix show samples of the three sets of images.

In this project, we hypothesize that the effectiveness of the Representative method is inversely proportional to the variation in colors between images in a set of images, as well as the variation in colors within each image. We anticipate that if there is significant variation in colors between images and within each image, the application of the Representative method will lead to a significant decrease in image quality and therefore, will not be appropriate. We expect the Human Black & White set to yield the highest quality reconstructed images due to its limited color variation, followed by the Carrot set which has a low color variation within images. We expect the Human set to yield the lowest-quality reconstructed images due to its significant color variation. However, these hypotheses need to be tested and validated through experimentation and analysis of the results.

2.3 Implementation Steps

The implementation consists of several steps:

1. The images are loaded into the environment and resized to 224x224 pixels. The images in the Human dataset are also converted to grayscale images during this process. At the end of this step, we have three sets of images: the Carrot set with images of few variations in colors, the Human set with images of many variations in colors, and the Human Black & White set with only shades of gray.

   We need to acknowledge the shape of the datasets as it greatly affects how we interpret the results. The Carrot and the Human sets have $224 \cdot 224 \cdot 3 = 150,528$ features and 200 observations. The reason why there are $224 \cdot 224 \cdot 3$ features is that the images are in the RGB color model with three values ranging from 0 to 255 representing each pixel, and there are 224x224 pixels. On the other hand, the Human Black & White set has $224 \cdot 224 = 50,176$ features because the images are in a grayscale color model with one value ranging from 0 to 255 representing each pixel.

2. In the second step, for each 50 images, we choose 1 representative image. Since there are 200 images in each dataset, we choose 4 representative images for each set. In the Carrot set

of images, each observation is an image. We run the K-means Clustering Algorithm on the Carrot set and use 4 images closest to the centers as 4 representative images. We conduct the same process for the remaining sets. Figure 9 in the Appendix shows the representative images that we choose.

3. The third step is to create codebooks from the representative images (see Figure 4). We run the K-means Clustering algorithm on each of the representative images with a chosen k. In this step, each RGB image is reshaped into each dataset of $224 \cdot 224$ observations and 3 features. On the other hand, each grayscale image is reshaped into each dataset of $224 \cdot 224$ observations and 1 feature. The K-means Clustering algorithm is run on each image. For each image, the result consists of the codebook (the indices and the corresponding prototype vectors) and the compressed image (the data representation of the image with only the indices of the prototype vectors).

4. Fourth step: we use the codebooks generated by the representative images to compress other images within the same cluster (see Figure 4). For example, if a picture, A, is in the same cluster with the representative image, R, then the codebook generated by R is used to compress A. In other words, the codebook generated by the representative image R is used to compress other images similar to R. In Figure 4, codebook 2 of image 2 is used to compress other similar images (all with the hand and a pile of carrots). The codebook is used to compress A by mapping each vector/pixel in A to the closest prototype vector in the codebook generated by R.

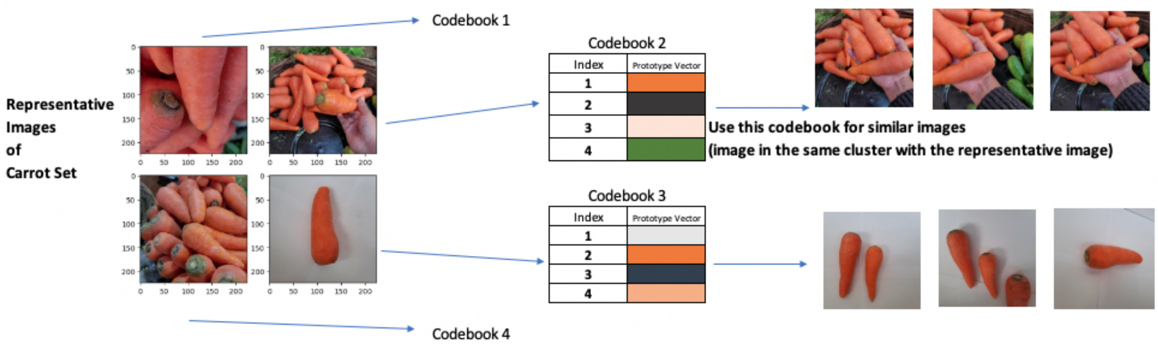

Figure 4: Illustrations of Implementation Step 2, 3, 4

5. In the fifth step, when we have all compressed images of three sets of images, we use an image quality metric, Mean Squared Error (MSE), to compare the quality of each compressed image to the original image. The MSE measures the average squared difference/distance between the original pixel values and the newly generated pixel values in the compressed images.

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_1(i,j) - I_2(i,j)]^2 \tag{1}$$

   where M and N are the height and width of the images, and $I_1(i,j)$ and $I_2(i,j)$ are the pixel values of the corresponding locations in two images. After having the MSE of all images, we compute the median MSE for each set. The reason why we choose to compute the median instead of the mean is that the distribution of the MSEs is right-skewed. In this case, the median represents the data better than the mean.

6. We repeat steps 3, 4, and 5 for each $k = 12, 24, 36, 48, 60$.

k	Human B&W	Carrot	Human
12	183.42	219.29	462.04
24	75.60	168.54	361.78
36	29.97	131.16	298.11
48	20.76	118.23	259.56
60	13.89	106.18	243.53

Table 1: Table of Median MSE across different k values

3 Results and Discussion

3.1 Results

The results of our experiments are summarized in Table 1, which shows the median MSE values obtained for each set of images and the value of k. Each pixel in the grayscale image is only represented by one value, while each pixel in the RGB image is represented by three values of red, green, and blue. To ensure consistency, the MSE of the grayscale image is multiplied by 3.

Our results show that the Human Black & White set has the lowest median MSEs across k values, which is consistent with our hypothesis. The Carrot set has the second-lowest median MSE, followed by the Human set, which has the highest median MSE. These results suggest that the effectiveness of the Representative method is indeed inversely proportional to the variation in colors between and within images in a set of images.

In addition, the median MSE decreases as k increases. This is consistent with the fact that as k increases, the compressed images are represented by more prototype vectors, or in other words, more colors. The table also shows a significant drop in the median MSE in the Human B&W set from $k = 24$ to $k = 36$. This indicates that the Representative works significantly better on the Human B&W set. It requires smaller k to achieve a great quality of reconstructed images.

Although MSE is a decent metric to measure image quality, human perception of quality may be different. Figure 5 shows the reconstructed image using a codebook generated by itself (left image) and the reconstructed image using a codebook generated by its representative image (right image). The left image has a greater quality than the right image. However, the right image is still of decent quality that can be used for tasks that do not require exact reconstruction.

Figure 5: Illustration of Image Quality

3.2 Future work

Several avenues for future work can be explored to build on the findings of this project. Since the project is only conducted on three sets of images, one possible direction is to investigate the effectiveness of the Representative method on additional sets of images with more diverse color variation. This will help determine if our finding in this paper is due to randomness. Another potential direction is to test out the other method, says the All method, which is to run the K-means Clustering Algorithm on each image and have a separate codebook for each image. We can then evaluate the runtime and image quality trade-off between the Representative and All methods. Finally, instead of using MSE, it may be valuable to use different image quality metrics or more human perception of quality to evaluate the effectiveness of these methods.

4 Conclusion

In conclusion, our findings support the hypothesis that the effectiveness of the Representative method is dependent on the degree of color variation within and between images. The Human Black & White set, which had the lowest degree of color variation, yielded the highest-quality compressed images, while the Human set, which had the highest degree of color variation, yielded the lowest-quality compressed images. The Carrot set, which had low color variation between images, fell in between in terms of compressed image quality.

5 Appendix

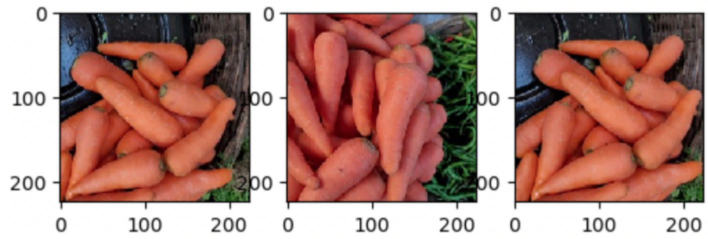

Figure 6: Samples of the Carrot set

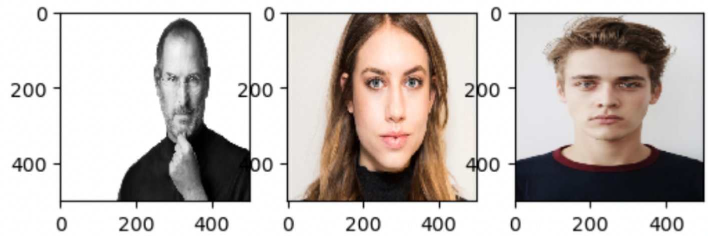

Figure 7: Samples of the Human set

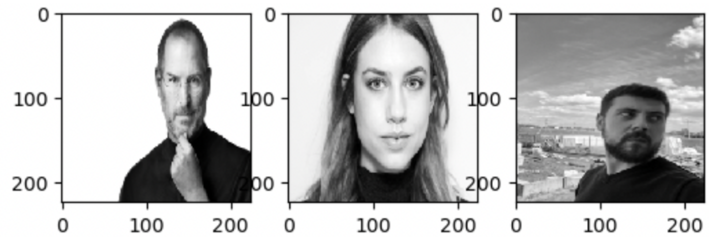

Figure 8: Samples of the Human Black & White set

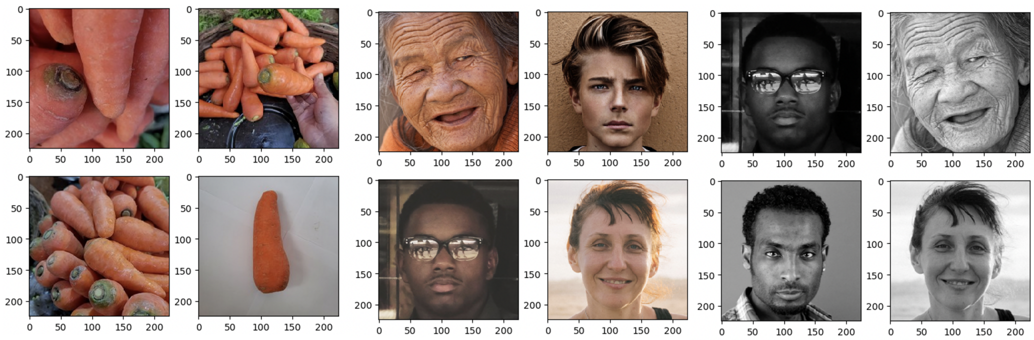

Figure 9: Representative images. From left to right, the first four images are the representative images of the Carrot set. The next are the representative images of the Human set. The last are the representative images of the Human B&W set.

References

[1] M. Israk Ahmed, Shahriyar Mamun, and Asif Asif. Dcnn-based vegetable image classification using transfer learning: A comparative study. pages 235–243, 05 2021.

[2] Dhruv Makwana. Face mask classification, Jul 2020.