CS339: Artificial Intelligence
Spring 2021

Image Reconstruction applying K-means: Color Comparison and Error Reduction

Amna Khalid

1. INTRODUCTION

Clustering algorithms are commonly used as a data exploration tool in Artificial Intelligence to get a better representation of the data. Clustering is defined as the identification of subgroups in the data such that the data points in one cluster are alike while data points in another cluster are not. We try to group the data points into clusters by finding the similarity measure between these points. Clustering is an unsupervised learning technique due to the fact that we don't have anything to compare the output of the clustering algorithm. We are only investigating the data by dividing it into distinguished subgroups.

There are different clustering algorithms and one of the most popular methods is the k-means clustering algorithm. K-means clustering algorithm is an unsupervised algorithm that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest cluster center, serving as a model of the cluster.

1.1 Vector quantization

Vector quantization is a method that allows the modeling of probability density functions by the distribution of prototype vectors. It works by dividing a large set of vectors into groups having roughly the same number of points closest to them. Each group is designed by its cluster's center point. The k-means algorithm takes the number of clusters, k, and a set of observation vectors to the cluster, as input. It returns the centers of the cluster for each of the k clusters. An observation is classified with the cluster closest to it.

A vector v is grouped to cluster i if it is closer to the center of cluster i than any other cluster's center. The k-means algorithm tries to minimize the distortion between the data, which is defined as the sum of the squared distances between each observation vector and its closest cluster center. This minimization is accomplished by iteratively reclassifying the observations into clusters and recalculating the centers of the clusters until a configuration is reached in which the centers of the clusters are stable.

Applying this idea, if an image has n grey levels, they can group these into k intervals, according to how close they are. Vector quantization thus can be performed on an image to change the representation of that image into something that is more meaningful and easier to analyze.

Thus for this research, using this process I will be examining how the optimal value of k changes for different images and how well the error is reduced for each value of k. Further, this research aims to explore whether images with a wider range of colors reach the optimal k value faster in comparison to images with a lower range of colors. For example, how does the optimal k value for an image with more range of colors differ from that of an image with only two colors i.e. black and white?

2. METHODOLOGY

To examine the optimal k values, vector quantization is performed on various images. Further, the sum of square error vs. k values graph is plotted for each image to get a visual representation of which k value is optimal for each image. This error is defined as the sum of the squared distance between each member of the cluster and the center of the cluster. These error representations will help determine the optimal number of clusters more efficiently. On these graphs, the value of k at the "elbow", i.e. the point after which the distortion/inertia starts decreasing in a linear fashion, is selected.

2.1 Image to data

Suppose we wish to compress a 24-bit color image. Each pixel is represented by one byte for each: red, blue, and green. A smaller 8-bit encoding is used since it reduces the amount of data by two-thirds. Running k-means with $k = 256$ will generate a table of 256 codes mapped to each cluster's center. This will fill up all potential 8-bit sequences. Instead of using a 3-byte value for each pixel, the 8-bit center of the cluster's index is used. The mapping is then returned in 8-bit codes and can be changed back to a 24-bit pixel value image.

To apply *k-means*, an array with as many rows as pixels and for each row/pixel, 3 columns, one for each color intensity (red, green, and blue) is needed. The image pixels are rearranged using this process, and an image is turned into an array to be processed by our k-means algorithm.

2.2 K-Means Algorithm

Let us consider an image with a resolution of $x * y$ and the image has to be cluster into k number of clusters. Let $f(x, y)$ be an input of pixels to the cluster and c_k be the cluster centers. The algorithm for k-means clustering will work as follows:

1. Initialize the number of clusters k and center.
2. For each pixel of an image, calculate the Euclidean distance d, between the center and each pixel of an image using:

$$d = \| f(x,y) - c_k \|$$

3. Assign all the pixels to the nearest center based on distance d.
4. After all the pixels have been assigned, recalculate the new position of the center.
5. Reshape the cluster pixels into an image.

2.3 Mean Sum of Squares

The basic idea behind cluster partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation (known as a mean variation or mean sum of square) is minimized:

$$minimize(\sum_{k=1}^{k} W(c_k))$$

Where $W(c_k)$ is defined as the within-cluster variation. The mean sum of squares measures the compactness of the clustering. The following algorithm is used to define the optimal clusters:

1. Compute k-means clustering for different values of k for each image.
2. For each k, calculate the mean sum of squares.
3. Plot the curve of the mean sum of square error according to the number of clusters k.

The location of a bend i.e “elbow” in the plot is recognized as an indicator of the appropriate number of clusters.

3. RESULTS

Experiments were conducted on a variety of nature images taken from Google. The sub-sections below show the detailed results of performing K-means clustering on these images.

3.1 Images and Colors

For this section, we focus on how the range of colors in different images can affect the visibility of the image. To demonstrate our results, we use the base value of k; $k = 2$. The results are shown as follows:

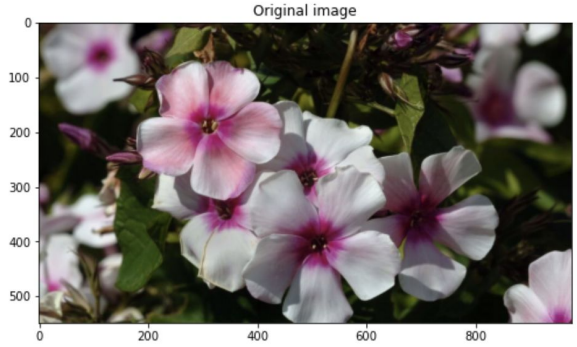

Fig. 1(a)

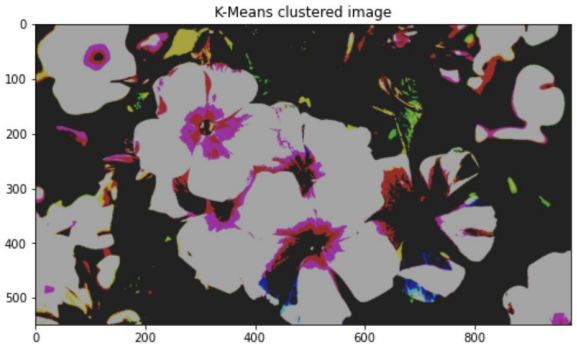

Fig. 1(b)

From figure 1(a) and 1(b), we see the reconstructed image using $k = 2$. Looking at the image, we are able to easily make out what the original image is. The same goes with figures 2 and 3 but we see figure 4 is not as easily recognizable in its reconstructed form.

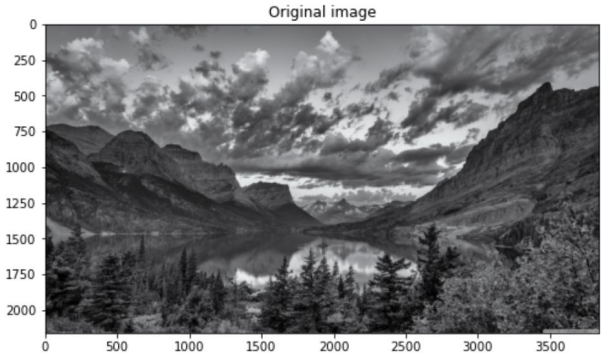

Fig. 2(a)

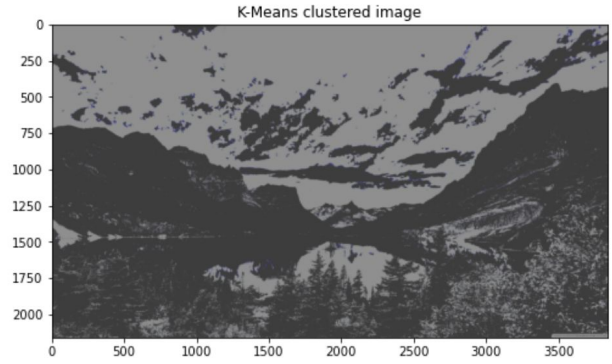

Fig. 2(b)

Looking at all the reconstructed images, it is also visible how figure 3 has the closest reconstruction from only two cluster centers. Further, the black and white images show to have a more recognizable reconstruction in comparison to colored images. Images that have more range of colors (for example fig. 1(a) has more colors compared to fig. 3(a)) also show to have more recognizable features while images that have fewer ranges of colors have less recognizable features.

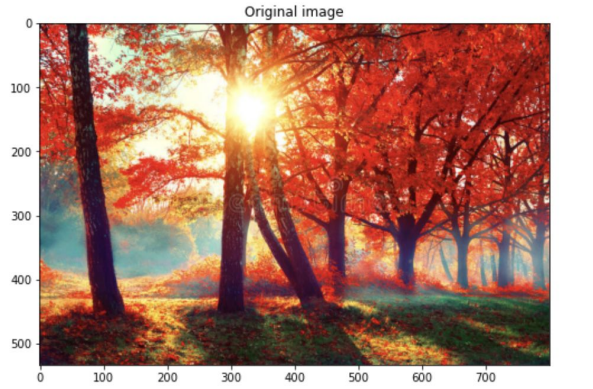

Fig. 3(a)

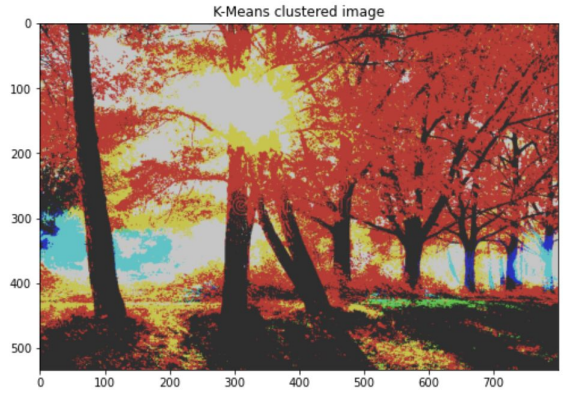

Fig. 3(b)

Fig. 4(a)

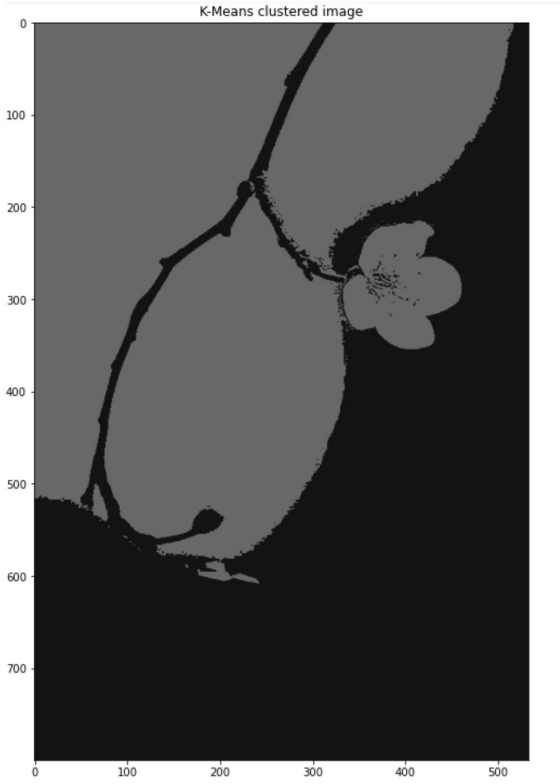

Fig. 4(b)

Analyzing just the black and white images, the K-means clustered images appear like a 2-D version of the original image. While there are a lot of details of the features encompassed in black and white reconstructed images, the depth however of the features is not as visible as it is in our colored examples.

3.2 Error Reduction

Computing error graphs for each image, we see that the curve is similar for each and every image that we looked at. We see the trend of error reducing while *k* values increase. The graphs produced are shown below.

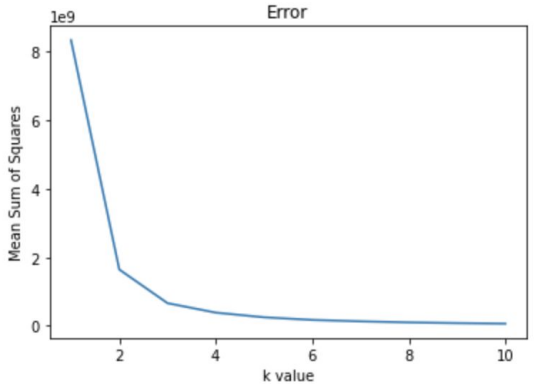

Fig. 5(a): Error graph for Fig. 1(a)

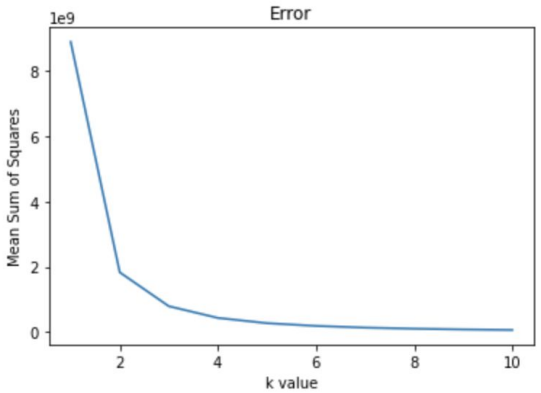

Fig. 5(b): Error graph for Fig. 4(a)

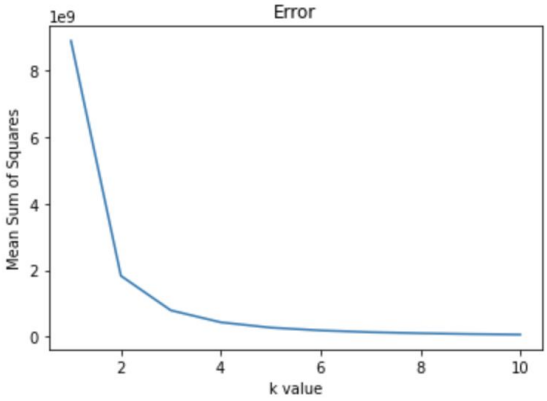

Fig. 5(c): Error graph for Fig. 3(a)

While the curve is similar in each graph, the mean sum of square (SSE) is larger for some images. Focusing on the x-axis, we see how the SSE has a range of up to 8 while the range for our black and white image is only up to 3.

4. CONCLUSION

The research has shown how the k-means clustering algorithm can recreate an image. The algorithm achieves this by creating new colors (clusters) which correspond to the centers of the clusters obtained with the *k-means* algorithm. The final centers are then the new colors that are then displayed in our recreated images.

The study shows that the error in between the clusters reduced as the value of k is increased. Furthermore, the k-means clustered images are shown to have less visibility due to the reduction of colors present. As the colors increase, the depth of the features in the image also becomes more visible. Where there are a variety of colors in an image, we see that the recreation of it by k-means is very similar and the image is recognizable with just $k = 2$.

With black and white images, the study shows that the image is not as recreated as the colored images. This is due to the low color varieties available in the images; which makes the k-means algorithm process most of the darker shades as black when we set a lower k value.

The study has shown how the k-means algorithm accomplishes reconstructing images. In conclusion, the results received were in accordance with how the algorithm operates.