Image Compression by K-Means Clustering

By

Asem Okby

Table Of Contents

Abstract

The internet is filled with data in the form of images. Storing those many images with their original size is expensive. Reducing the size of the images will allow us to use the space wisely and even store more images. In this project, we tackle the image compression problem using K-Means Clustering. It is a clustering algorithm that clusters the given data points to k clusters. K-Means's aim is to optimize the position of the cluster centroids, which minimizes the cost function used e.g. sum of squared distances. K-Means works in an iterative way. First, it starts with randomly assigned cluster centroids and iteratively optimizes the position of the centroids. K-means stops when it converges or when the maximum number of iterations is reached. Besides, we interpret our results and use different metrics to choose the best possible value of K.

Discovering The Data

We start off with understanding, visualizing, and learning some statistics about our data.

Visualizing the Images

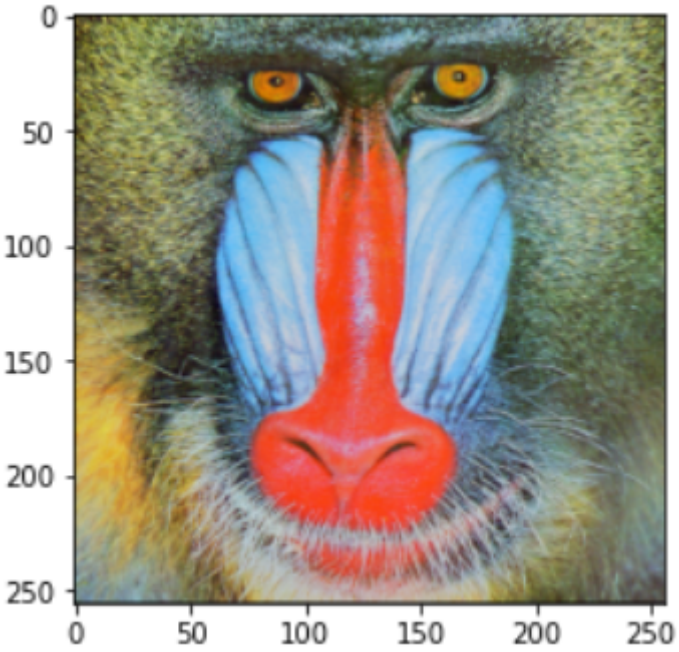


Image1 - Baboon

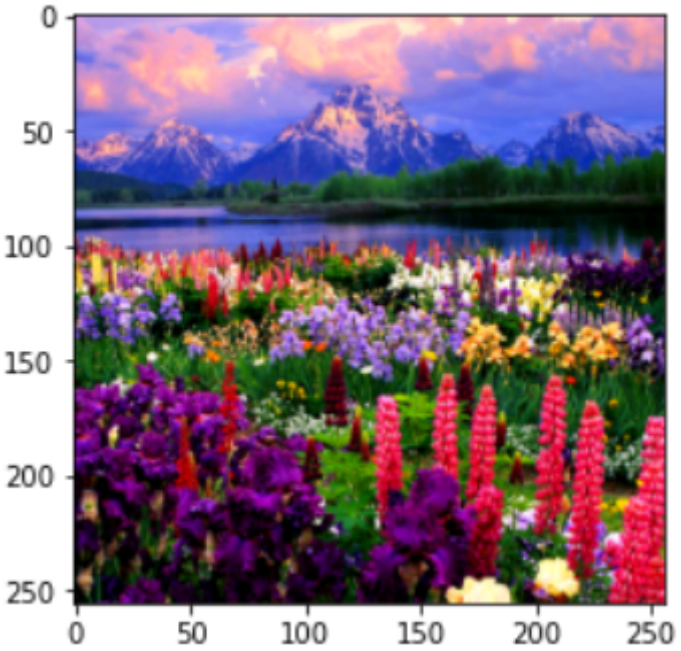


Image 2 - Flowers

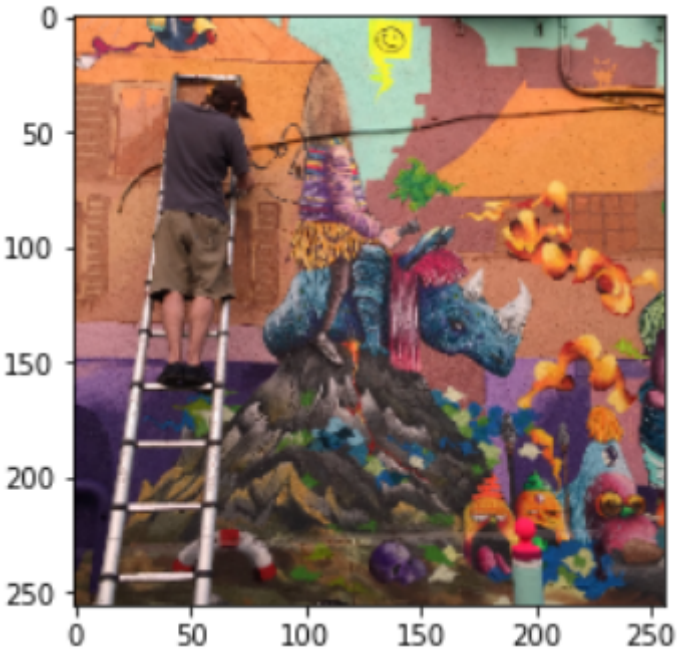


Image 3 - Graffiti

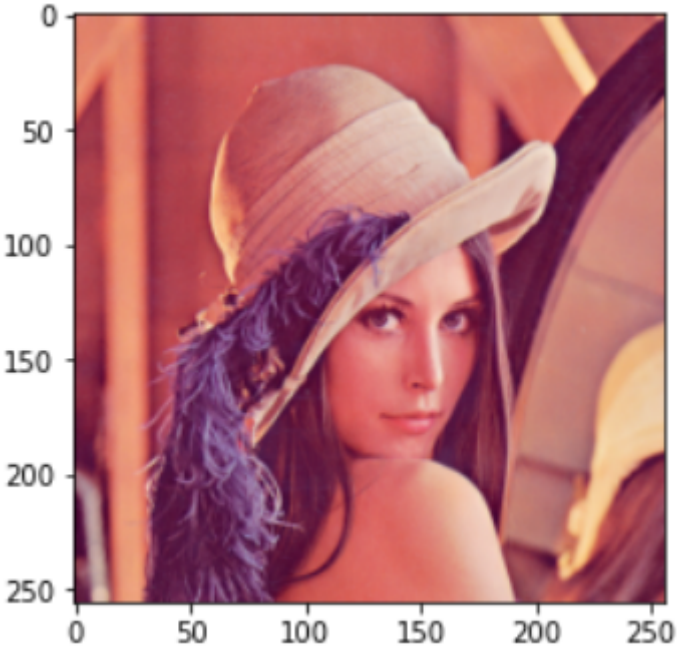


Image 4 - Lena

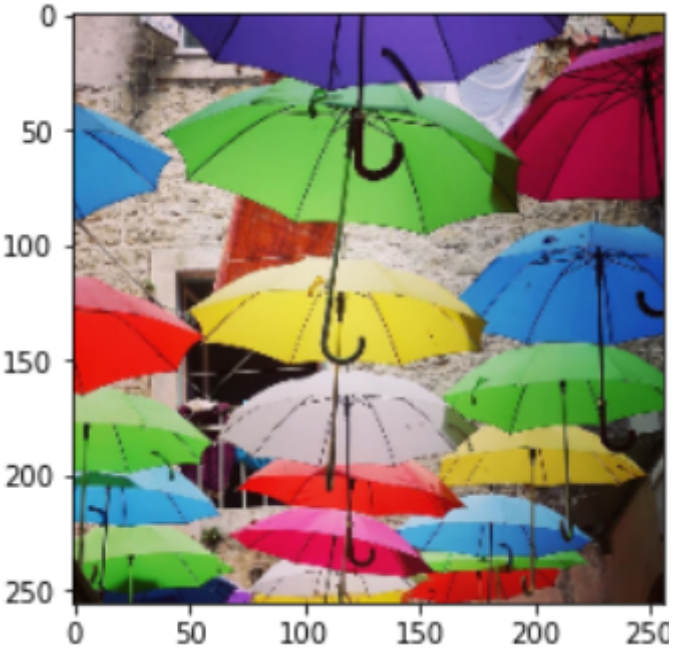


Image 5 - Umbrella

Size of the Images

The following are the Images' names and their corresponding size in bytes.

Image	Size (Bytes)
baboon	651142
flowers	615128
graffiti	1075269
lena	473831
umbrella	279534

Number of Unique Colors

The following are the Images' names and the number of unique colors in each.

Image	# of unique colors
baboon	62070
flowers	57848
graffiti	47091
lena	48331
umbrella	49461

Modelling

Compressing the Images

We compress or cluster each image with different K values from 1 to 8. In the notebook, we visualize the compressed images, 8 images for each image. We also calculate the size, in bytes, of the compressed images. Besides, for each K-Means model we train, we calculate the following metrics: WCSS: Within Cluster Sum of Squares, BCSS: Between Cluster Sum of Squares, and Explained Variance (Silhouette Coefficients).

Visualizing Compressed Images

Here, we visualize the compressed images for each value of K.

Image 1 - Baboon

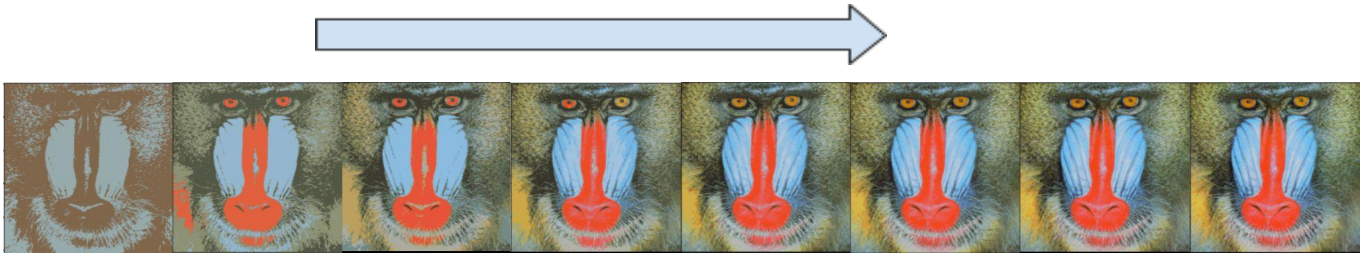


Image 2 - Flowers

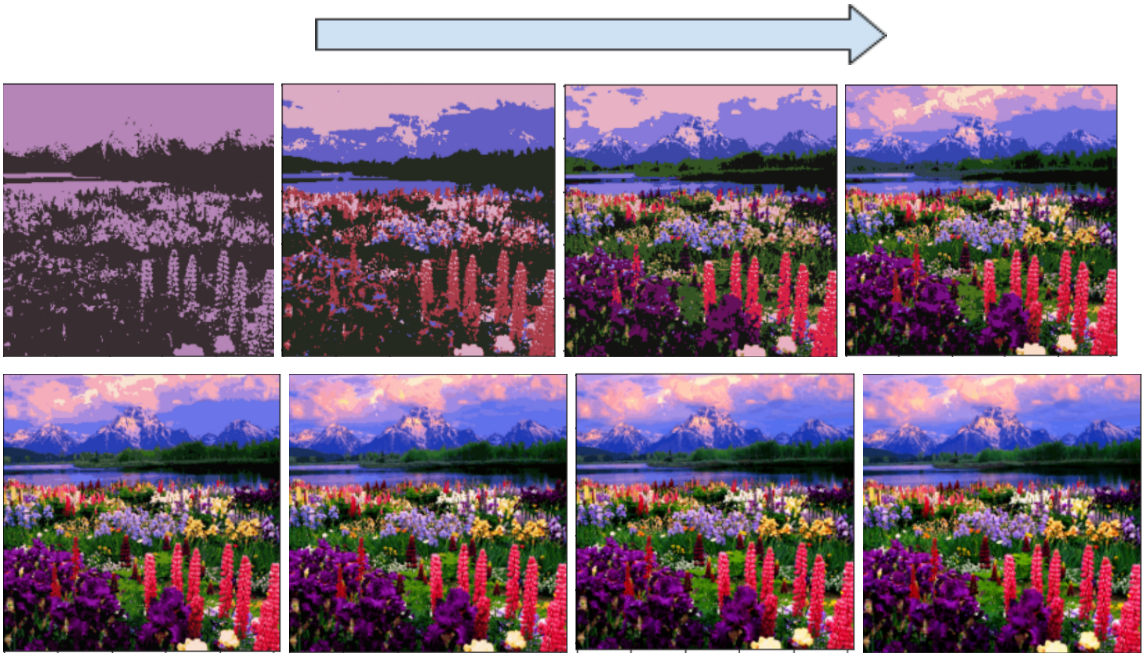


Image 3 - Graffiti

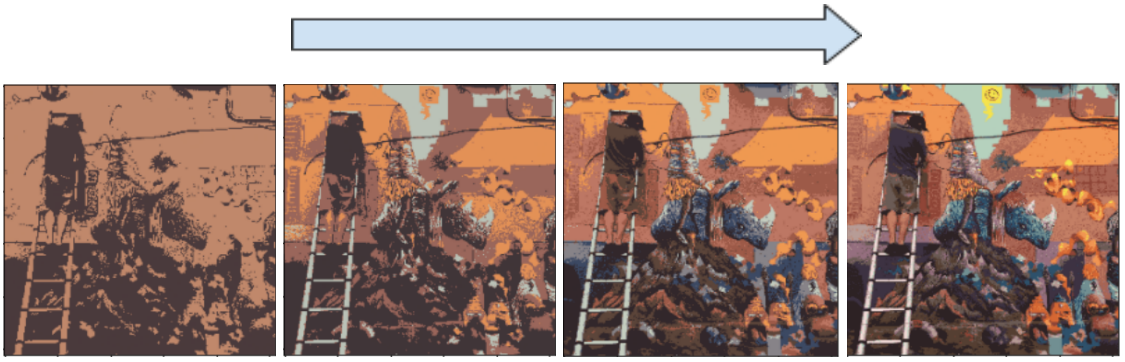
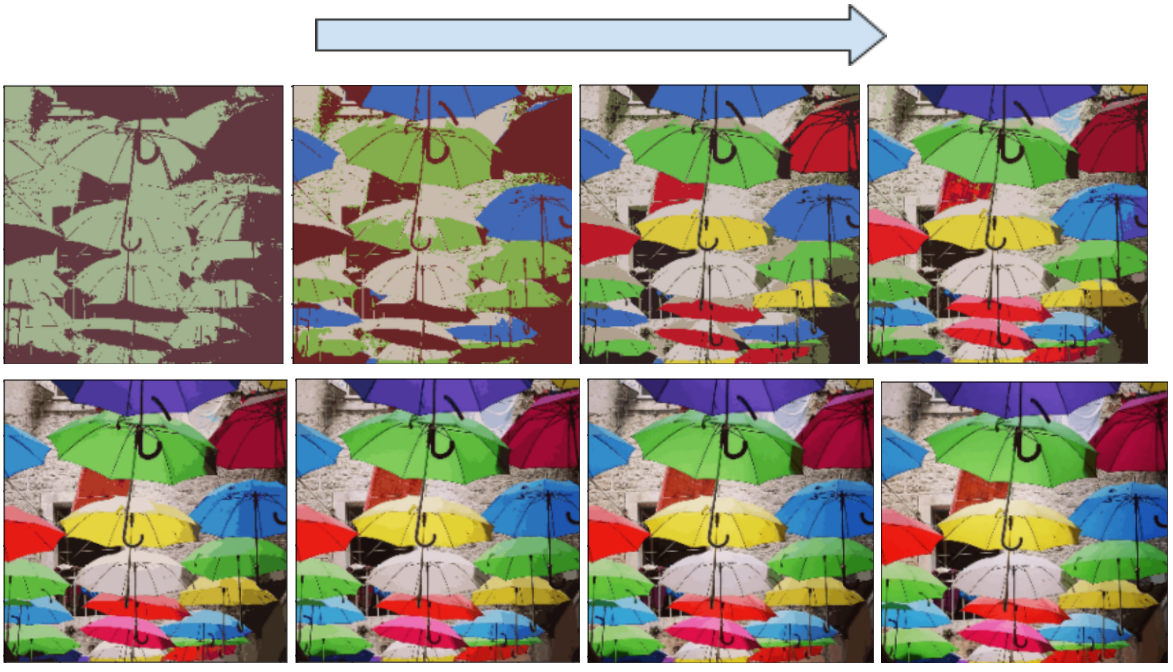


Image 4 - Lena

Image 5 - Umbrella



Results

Here, we display the previous results in a compact way, in tables.

Image 1 - Baboon

# of Clusters	Centroid Colors	WCSS	BCSS	Explained Variance	Size (Bytes)
2	[dimgray, darkgray]	3.036815e+08	2.308232e+08	0.387091	10574
4	[darkslategray, lightslategray, royalblue, tan]	1.233222e+08	4.112292e+08	0.426190	16499
8	[gray, royalblue, darkslategray, burlywood, slategray, cadetblue, dimgray, darkgray]	6.329426e+07	4.710088e+08	0.366709	28989
16	[dimgray, mediumpurple, darkseagreen, darkslategray, royalblue, darkslategray, gray, b...	3.426829e+07	4.996957e+08	0.326259	45594
32	[royalblue, slategray, silver, darkslategray, darkgray, mediumaquamarine, burlywood, d...	1.942369e+07	5.148377e+08	0.301529	69740
64	[darkslategray, darkgray, darkslateblue, burlywood, slategray, slateblue, royalblue, d...	1.176883e+07	5.224881e+08	0.265358	98542
128	[royalblue, dimgray, tan, darkseagreen, darkslategray, gray, steelblue, darkgray, sand...	7.451718e+06	5.268100e+08	0.246431	124274
256	[gray, royalblue, darkslategray, tan, steelblue, darkslategray, cadetblue, cadetblue, ...	4.724506e+06	5.295759e+08	0.235796	140838

Table 1 - Results of Image 1

Image 2 - Flowers

# of Clusters	Centroid Colors	WCSS	BCSS	Explained Variance	Size (Bytes)
2	[darkslategray, rosybrown]	4.883497e+08	6.564083e+08	0.493603	8490
4	[darkslategray, indianred, plum, darkslateblue]	2.575739e+08	8.871692e+08	0.451220	15386
8	[black, palevioletred, darkslategray, indigo, slateblue, steelblue, thistle, sienna]	1.394658e+08	1.005208e+09	0.390555	27832
16	[plum, darkgreen, sienna, mediumblue, black, lightcoral, slategray, midnightblue, lave...	7.565526e+07	1.069110e+09	0.366272	44421
32	[black, plum, purple, dodgerblue, indianred, lightsteelblue, saddlebrown, forestgreen,...	4.260042e+07	1.101986e+09	0.338151	67643
64	[palevioletred, darkslategray, paleturquoise, darkslateblue, black, plum, firebrick, d...	2.519309e+07	1.119052e+09	0.306412	89185
128	[black, lightslategray, plum, brown, slateblue, indianred, darkslategray, lavender, ro...	1.529014e+07	1.129234e+09	0.288189	108955
256	[lightsteelblue, black, indianred, darkslateblue, darkslategray, darkgray, saddlebrown...	9.332286e+06	1.135201e+09	0.275090	124170

Table 2 - Results of Image 2

Image 3 - Graffiti

# of Clusters	Centroid Colors	WCSS	BCSS	Explained Variance	Size (Bytes)
2	[steelblue, darkslategray]	2.331273e+08	3.603836e+08	0.517551	8843
4	[cornflowerblue, darkslategray, slategray, silver]	1.154685e+08	4.779705e+08	0.457149	16164
8	[darkslategray, steelblue, silver, darkslategray, gray, cornflowerblue, darkslateblue,...	6.473189e+07	5.289821e+08	0.365531	29821
16	[slategray, darkslategray, steelblue, dimgray, darkgray, cornflowerblue, darkolivegree...	3.846578e+07	5.551325e+08	0.345935	45557
32	[midnightblue, slategray, silver, cornflowerblue, dimgray, gray, darkslategray, darksl...	2.154709e+07	5.719095e+08	0.333193	62929
64	[slategray, darkslategray, royalblue, silver, steelblue, dimgray, cornflowerblue, stee...	1.180740e+07	5.815925e+08	0.327650	80733
128	[darkslateblue, darkgray, black, cornflowerblue, darkslategray, lavender, cornflowerbl...	6.693995e+06	5.867930e+08	0.314061	98757
256	[cornflowerblue, darkslategray, silver, slateblue, darkslategray, darkolivegreen, dark...	3.962399e+06	5.895112e+08	0.291554	115624

Table 3 - Results of Image 3

Image 4 - Lena

# of Clusters	Centroid Colors	WCSS	BCSS	Explained Variance	Size (Bytes)
2	[mediumpurple, darkslateblue]	1.469486e+08	2.600167e+08	0.537850	5371
4	[lightsteelblue, slateblue, mediumslateblue, indigo]	5.153244e+07	3.558647e+08	0.482217	11320
8	[darkslateblue, mediumpurple, indigo, lightsteelblue, slateblue, cornflowerblue, slate...	2.452611e+07	3.826131e+08	0.412836	19610
16	[lightsteelblue, darkslateblue, slateblue, midnightblue, gray, lightsteelblue, slatebl...	1.229785e+07	3.949539e+08	0.383749	29430
32	[darkslateblue, mediumpurple, midnightblue, slateblue, lightblue, slateblue, dimgray, ...	6.590001e+06	4.004453e+08	0.328287	45204
64	[mediumpurple, darkslateblue, lightsteelblue, slateblue, cornflowerblue, darkslateblue...	3.847780e+06	4.032377e+08	0.291699	64144
128	[darkslateblue, cornflowerblue, slateblue, midnightblue, lightsteelblue, darkslateblue...	2.383014e+06	4.046877e+08	0.262127	84053
256	[slateblue, cornflowerblue, indigo, skyblue, slateblue, darkslateblue, darkslateblue, ...	1.496036e+06	4.056131e+08	0.244789	99407

Table 4 - Results of Image 4

Image 5 - Umbrella

# of Clusters	Centroid Colors	WCSS	BCSS	Explained Variance	Size (Bytes)
2	[darkslategray, darkseagreen]	5.394247e+08	4.170130e+08	0.388216	7415
4	[sienna, midnightblue, silver, mediumseagreen]	2.808022e+08	6.762545e+08	0.438961	11694
8	[mediumturquoise, mediumblue, silver, peru, darkslategray, mediumseagreen, darkslategr...	1.194689e+08	8.370105e+08	0.458912	18277
16	[sienna, lightgray, darkslategray, mediumblue, mediumseagreen, black, mediumturquoise,...	5.765479e+07	8.989985e+08	0.458813	26125
32	[blue, slategray, darkslategray, peru, tan, mediumseagreen, brown, turquoise, darkslat...	2.825401e+07	9.284798e+08	0.418305	38945
64	[black, skyblue, darkslateblue, peru, blue, mediumseagreen, indigo, lightgray, indianr...	1.419062e+07	9.420856e+08	0.396013	52992
128	[thistle, saddlebrown, darkslateblue, seagreen, indigo, sienna, darkturquoise, black, ...	7.759495e+06	9.488064e+08	0.353285	70465
256	[darkseagreen, indigo, sienna, mediumseagreen, darkslategray, lightsteelblue, mediumtu...	4.521749e+06	9.520437e+08	0.317892	85140

Table 5 - Results of Image 5

Interpreting the Results

Optimal Elbows

In order to choose the optimal value of K, we use Optimal Elbows. The point of the elbow or inflection is the best value of K. We can create optimal elbows using the inertia, which is the WCSS score or distortions. Since we already calculated the WCSS scores, We use them to create the optimal elbows.

Image 1 - Baboon

In Figure 1, the elbow or the point of the inflection seems to be at the value K = 4, meaning 16 clusters (2^4).

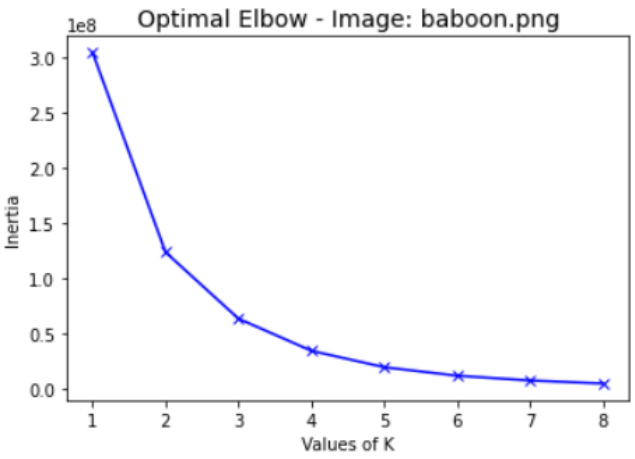


Figure 1 - Optimal Elbow for Image 1

Image 2 - Flowers

In Figure 2, the elbow or the point of the inflection seems to be at the value K = 4 meaning 16 clusters (2^4).

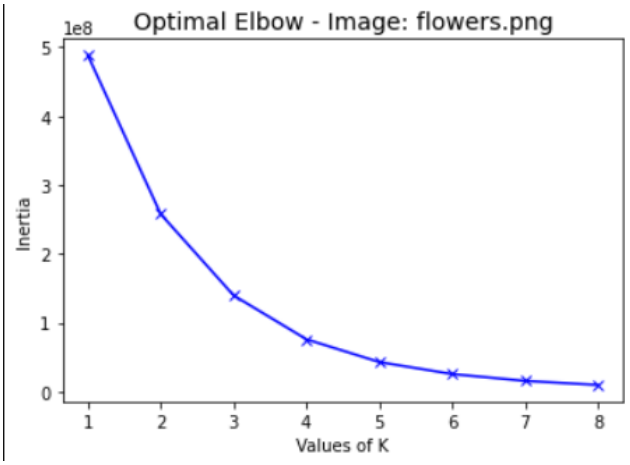


Figure 2 - Optimal Elbow for Image 2

Image 3 - Graffiti

In Figure 3, the elbow or the point of the inflection seems to be at the value K = 4 meaning 16 clusters (2^4). Even 5 could be seen here as the optimal value.

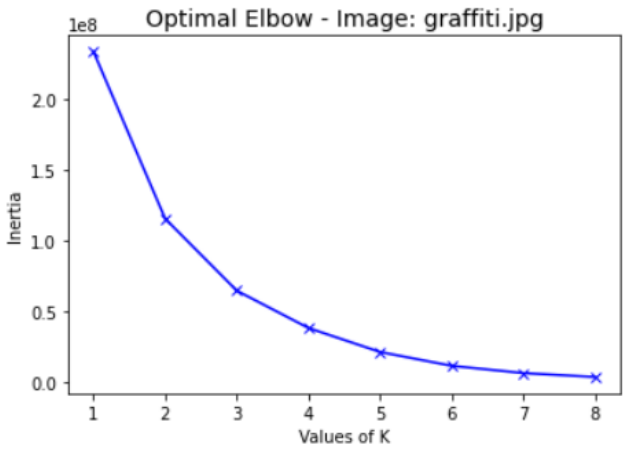


Figure 3 - Optimal Elbow for Image 3

Image 4 - Lena

In Figure 4, the elbow or the point of the inflection seems to be at the value K = 4 meaning 16 clusters (2^4).

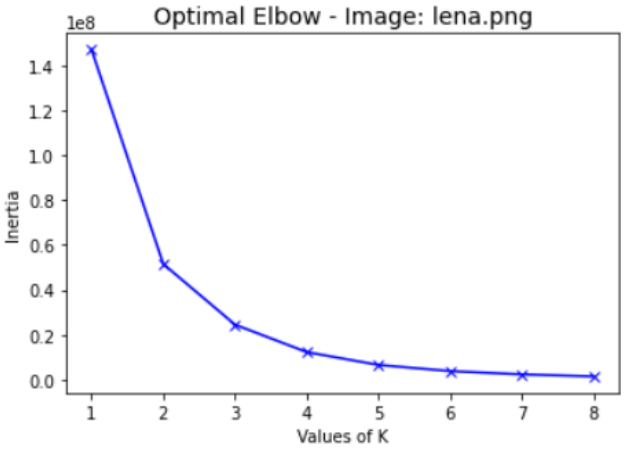


Figure 4 - Optimal Elbow for Image 4

Image 5 - Umbrella

In Figure 5, the elbow or the point of the inflection seems to be at the value K = 4 meaning 16 clusters (2^4).

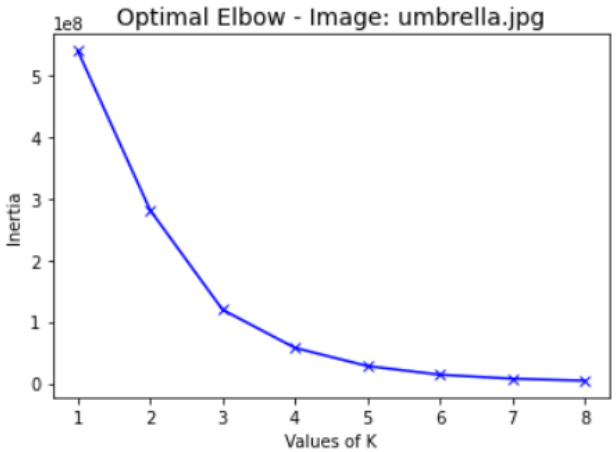


Figure 5 - Optimal Elbow for Image 5

Trade-off: Explained Variance and Image Size

Here we show another way of choosing the value of K, in other words, the number of clusters while considering the trade-off between the explained variance and image size.

Silhouette analysis is used for understanding the separation distance between the clusters. The Explained Variance (Silhouette coefficient) gives us an idea about the distance between a sample and the neighboring clusters. It has a range of [-1, 1]. Therefore, a value close to 1 indicates that the sample is far away from the neighbouring clusters. A value of 0 indicates that the sample is very close to the neighboring clusters and those samples might have been assigned to the wrong cluster.

Image 1 - Banoon

As you can see, in Figure 5, we see the negative correlation between the explained variance and the Image size. Our goal is to choose the best quality image with the least memory requirement possible. The graph shows that at the value K = 2, the explained variance is at its highest. Therefore setting K as 2 seems to be a reasonable decision. Also, notice at K = 1, the size is indeed smaller than that of the one at K = 2, however, the quality is not as good as that of the one at K = 2.

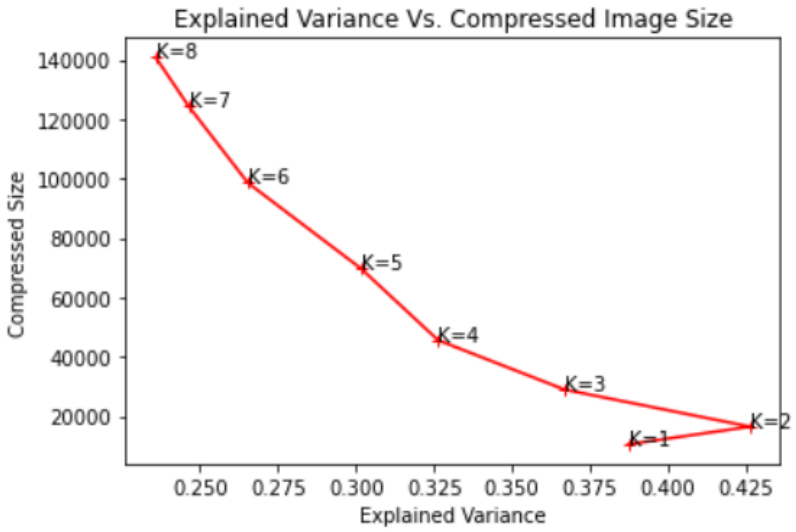


Figure 5

Image 2 - Flowers

Here, in Figure 6, we may say directly that the best value of K is 1, however, the value at K = 3 seems to be better, and it is even close to the optimal elbow agreed on. Also, the point at K = 3 is the point of inflection, elbow, showing that the difference between values of K past that is not that great.

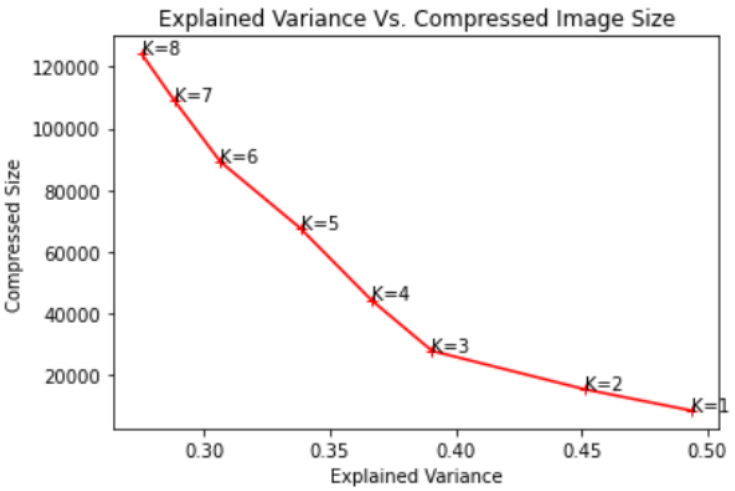


Figure 6

Image 3 - Graffiti

Again, in Figure 7, we might be tempted to go with K = 1, however, a trade-off lives at the point K = 3. Also, this is close to the value the optimal elbow shows.

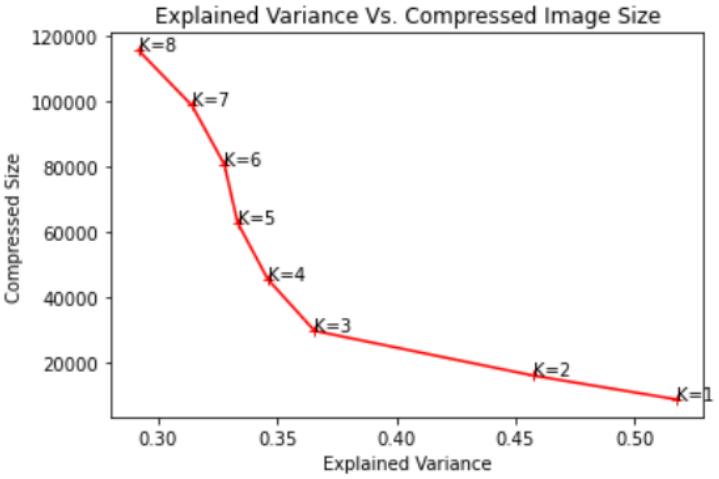


Figure 7

Image 4 - Lena

Figure 8, shows again, the same trend we saw previously. That is the inflection point is staying at K = 3. Again, here it is reasonable to choose the best value as K = 3.

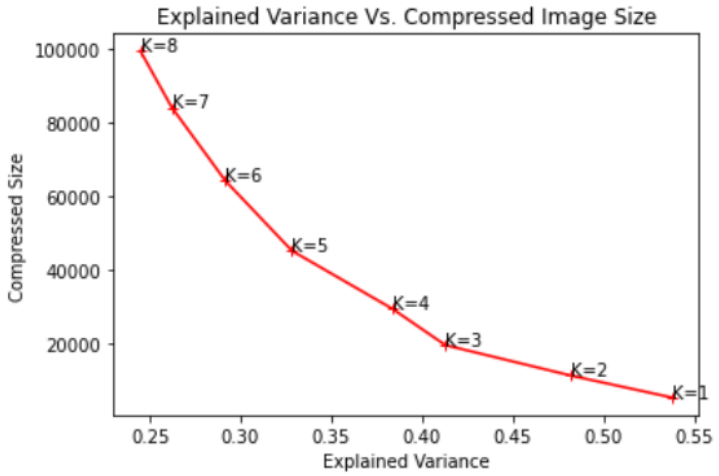


Figure 8

Image 5 - Umbrella

This graph, Figure 9, is more interesting. The best quality ever is at K = 3 or K = 4. And the best size is at K = 1, but with a bad quality. Certainly, choosing here K = 4 or K = 3 for the quality makes sense. However, the size for each is different. K = 3 is a better choice since it delivers the best quality with the least memory requirement.

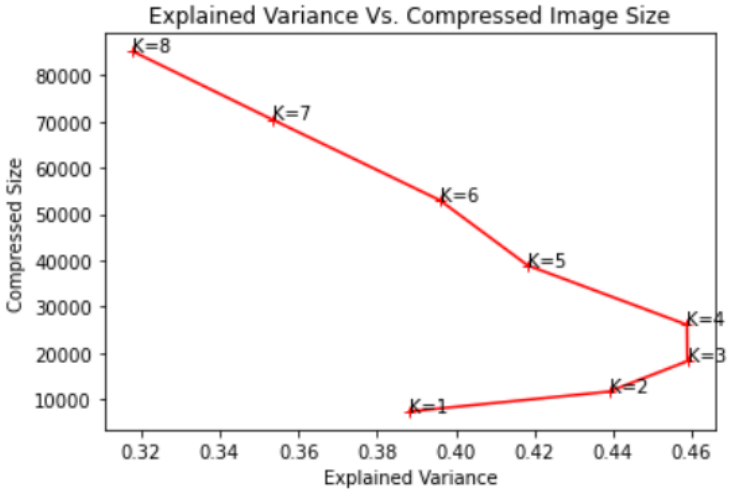


Figure 9

Original Vs. Best Quality

Finally, we show here, in Figure 10, a comparison between the original images and the best quality compressed image.

Image Name	Original Size	Compressed Size	Explained Variance
baboon.png	651142	16499	0.426190
flowers.png	615128	27832	0.390555
graffiti.jpg	1075269	29821	0.365531
lena.png	473831	19610	0.412836
umbrella.jpg	279534	18277	0.458912

Figure 10

References

[1] Lloyd, Stuart P. "Least squares quantization in PCM." Information Theory, IEEE Transactions on 28.2 (1982): 129-137.