Chris Schulz
Jake Graham
PA3

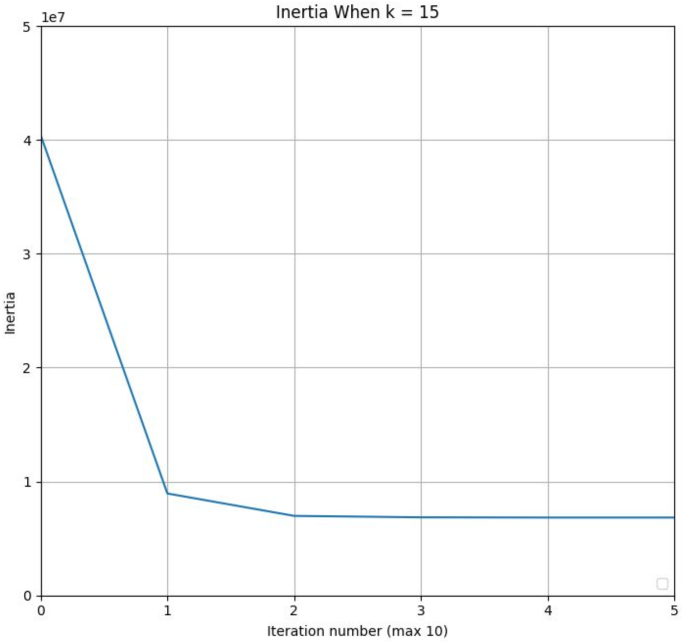
Clustering Assignment

Introduction:

For this project we created a kMeans model to compress images using clustering. First step is to define the number of clusters, then calculate the centroid and its color, then replace all the other pixels in that cluster to have that color. This creates a new “compressed” image that has “k” colors and simplifies the palette of the image. Below is the requested responses to various guidelines we were provided.

Results:

1. For one of the images that have been supplied to you, run kmeans 10 times with k = 15 and report/plot the sum of squared errors (inertia). Briefly explain why the results vary (1 or 2 sentences).

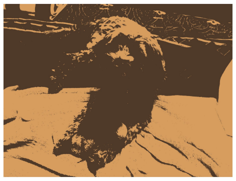


a) For this question we decided to run kmeans on Image 2, what we saw was that the sum of squared errors was very high initially and then decreases as the iterations increase. This is because the centroids update their locations in order to try and minimize the sum of squared errors.

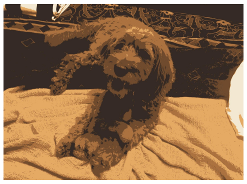
2. For each of the 4 images, perform K-Means clustering with the following values for k: 2, 5, 10, 15, 20. For each value, report the number of iterations it took for the algorithm to converge as well as the inertia value in a table or as a plot (a total of 20 runs).
    a) Image 1

Original image

Compressed versions of Image 1



Compressed with k = 2



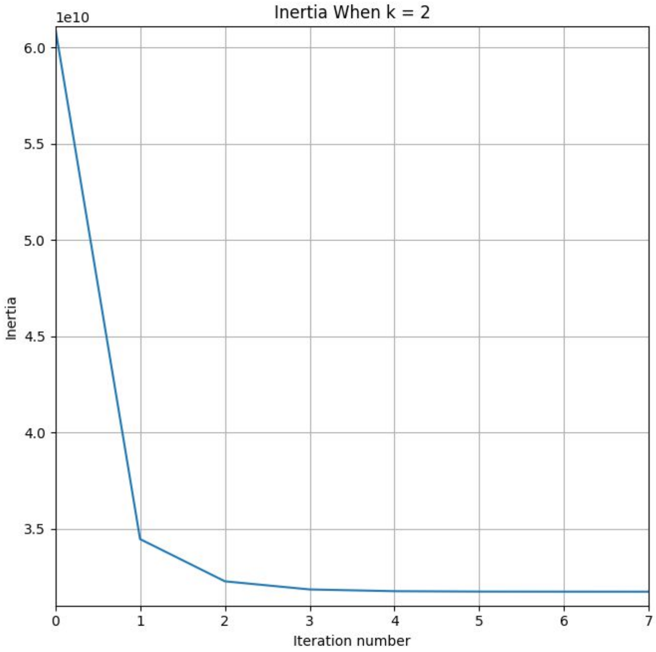
Compressed with k = 5

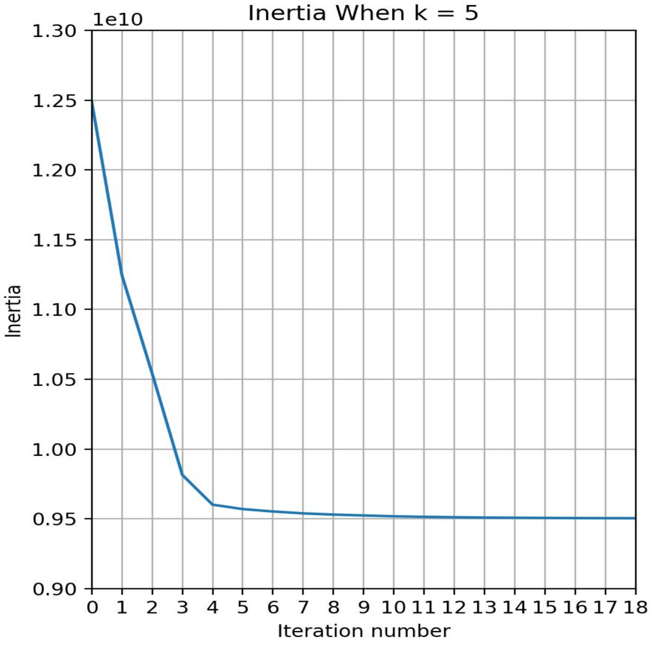
Compressed with k = 10

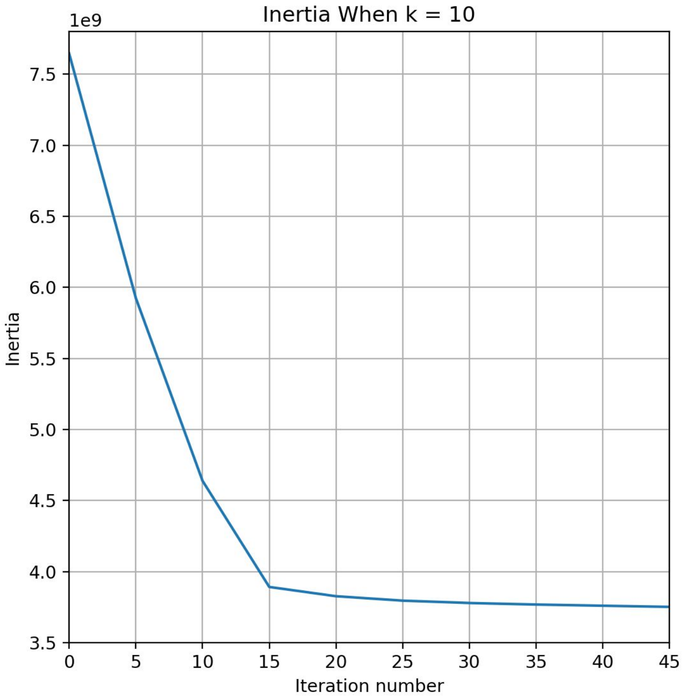
Compressed with k = 15

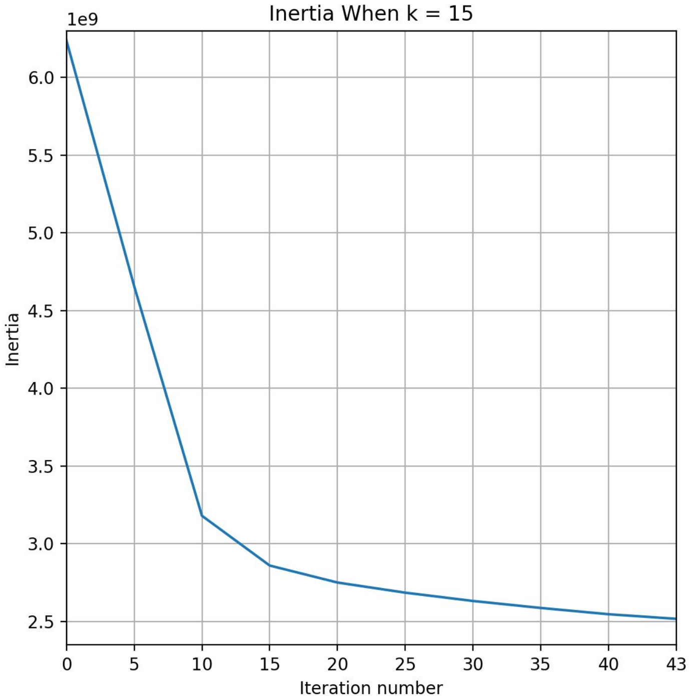
Compressed with k = 20

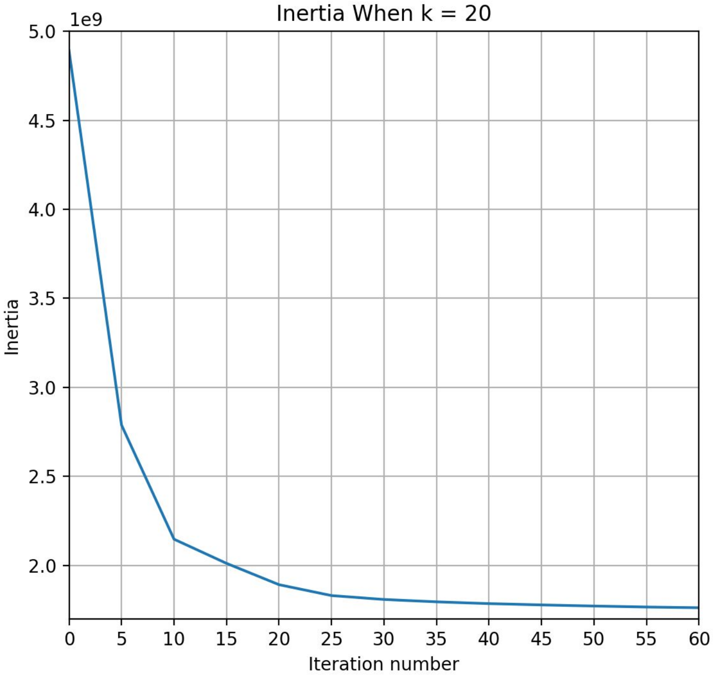
Inertia values over each iteration (to convergence) for each compressed image 1



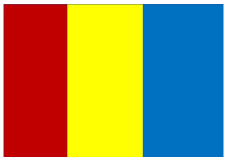



Inertia When k = 10


Inertia When k = 15

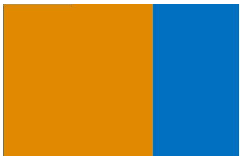

Inertia When k = 20

b) Image 2

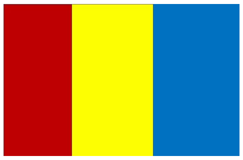


Original image

Compressed versions of image 2



Compressed with k = 2



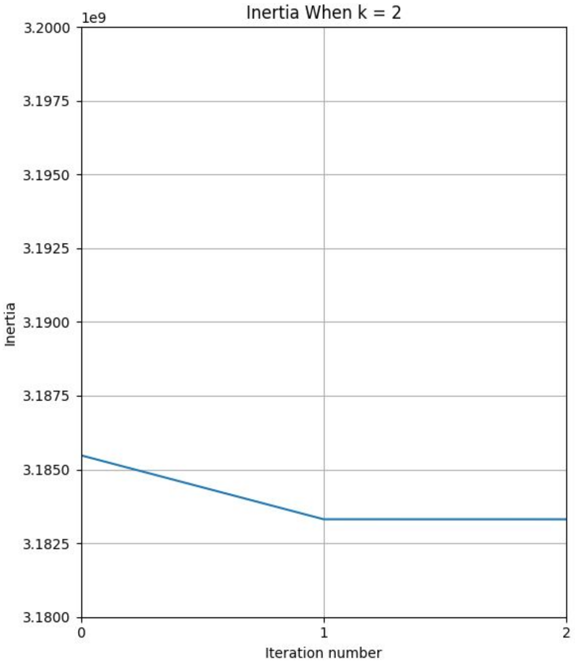
Compressed with k = 5

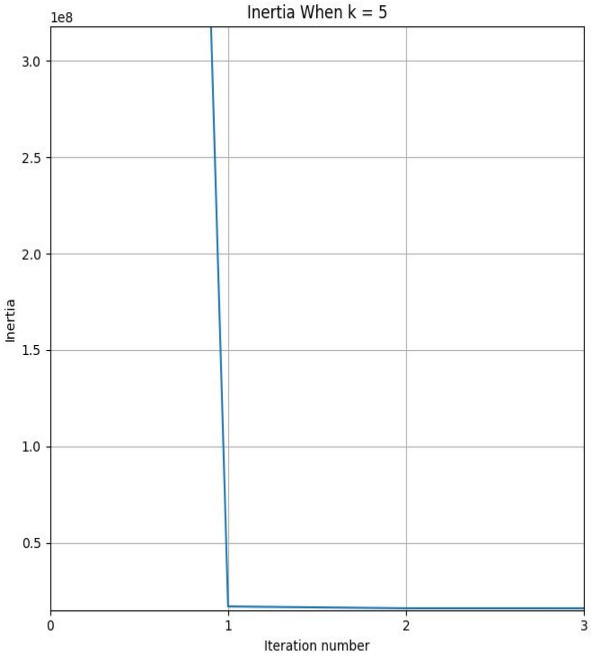
Compressed with k = 10

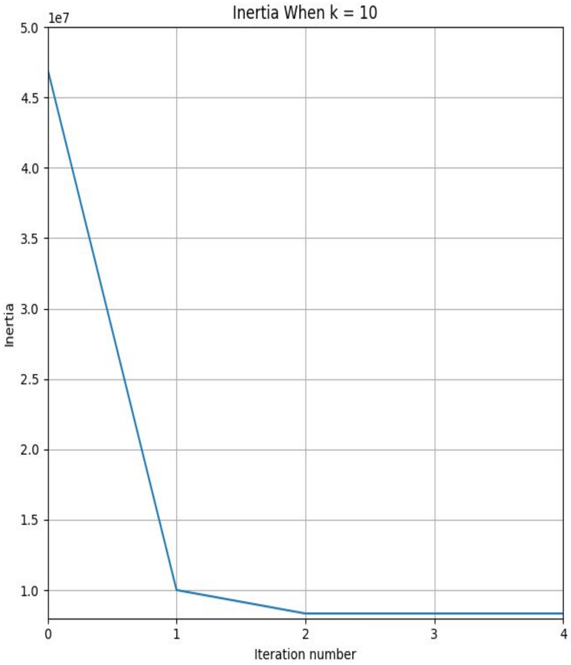
Compressed with k = 15

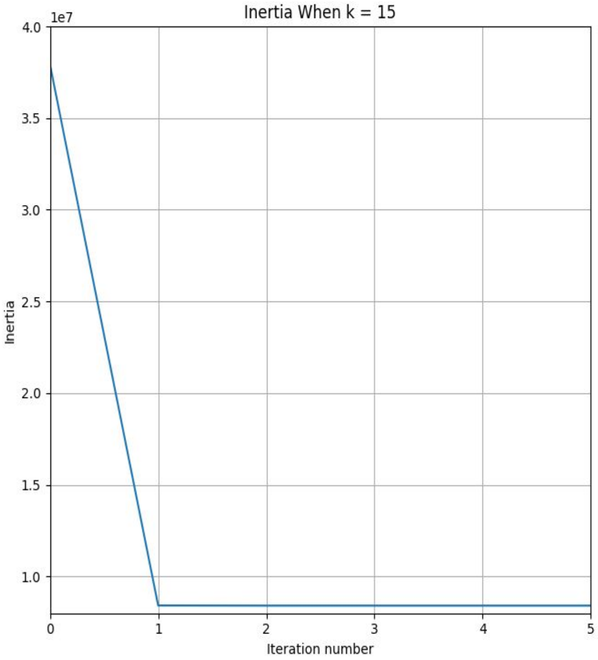
Compressed with k = 20

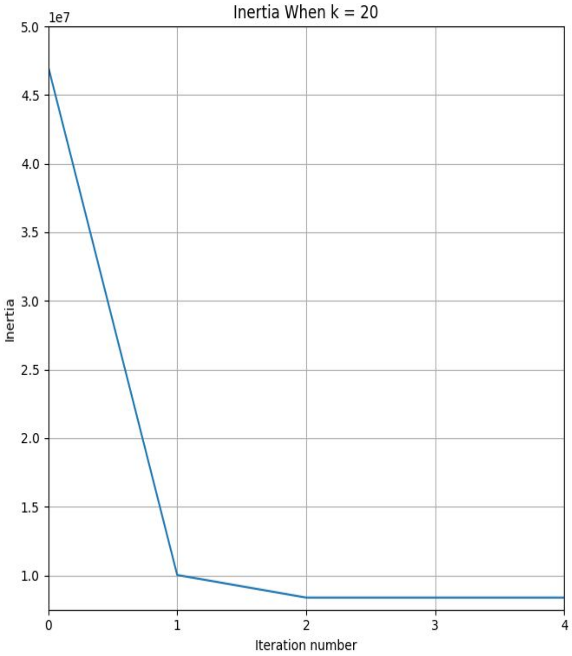
Inertia values over each iteration for each compressed image 2



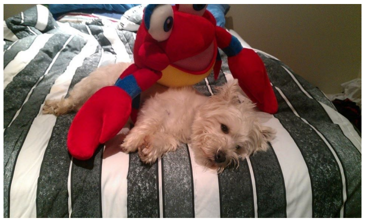






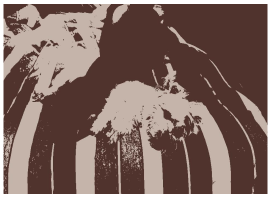


c) Image 3

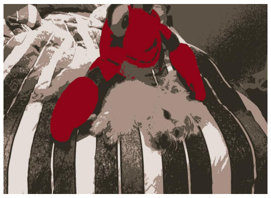


Original Image

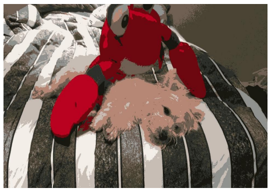
Compressed versions of image 3



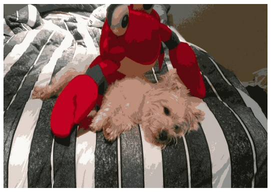
Compressed with k = 2



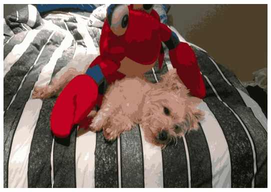
Compressed with k = 5



Compressed with k = 10

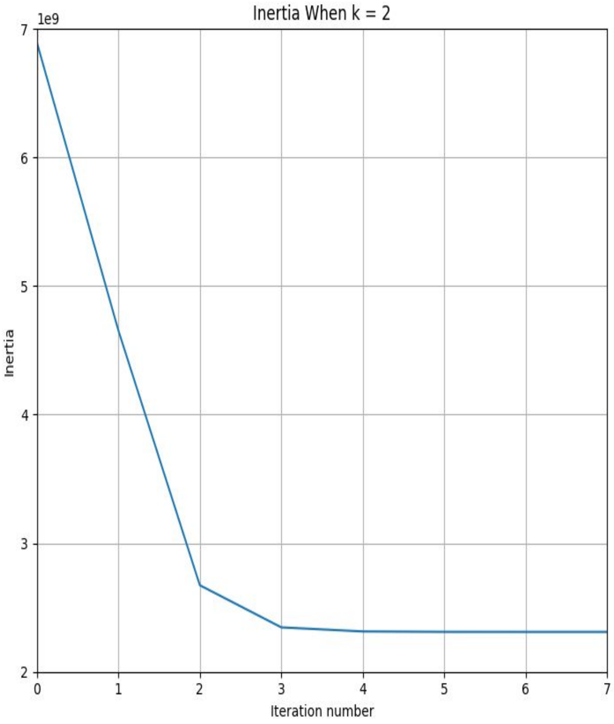


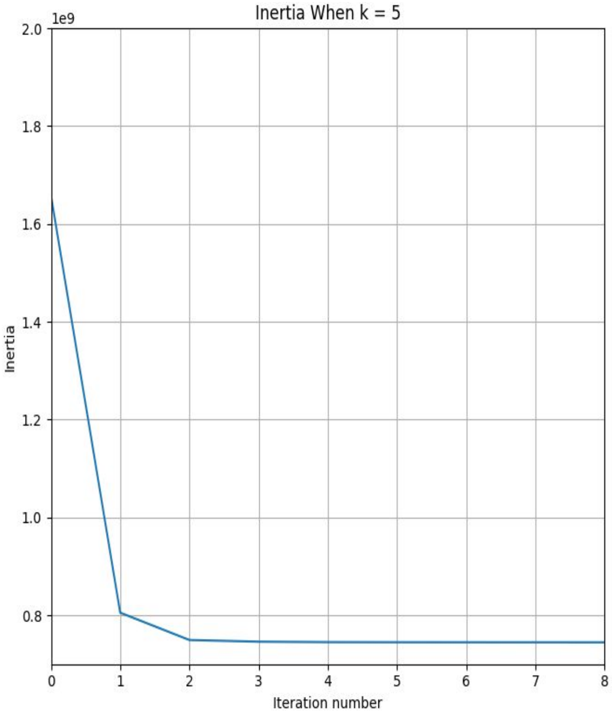
Compressed with k = 15

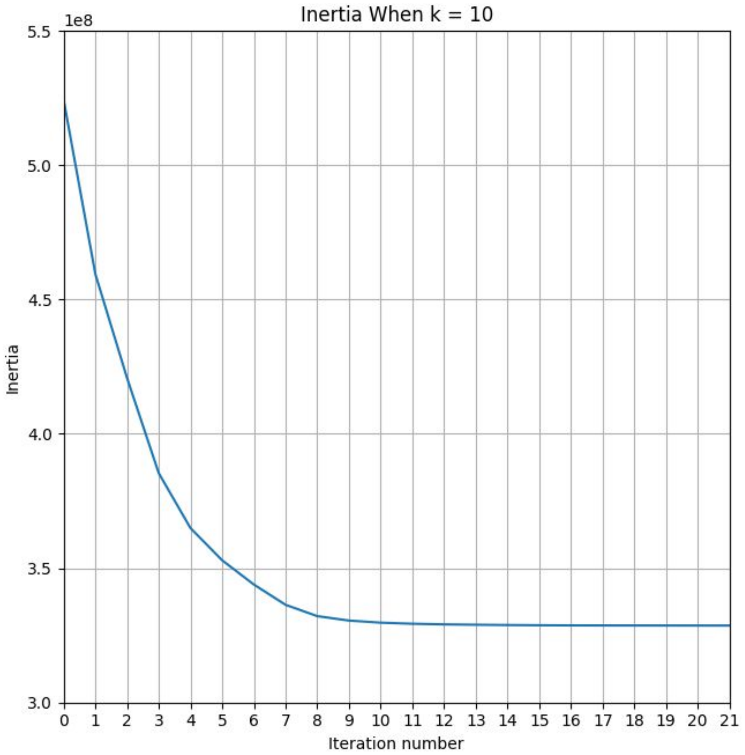


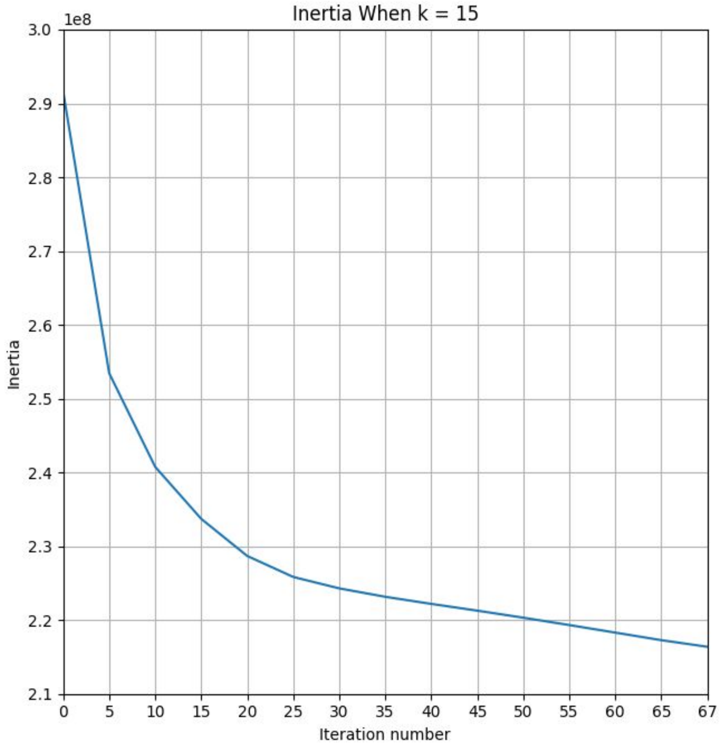
Compressed with k = 20

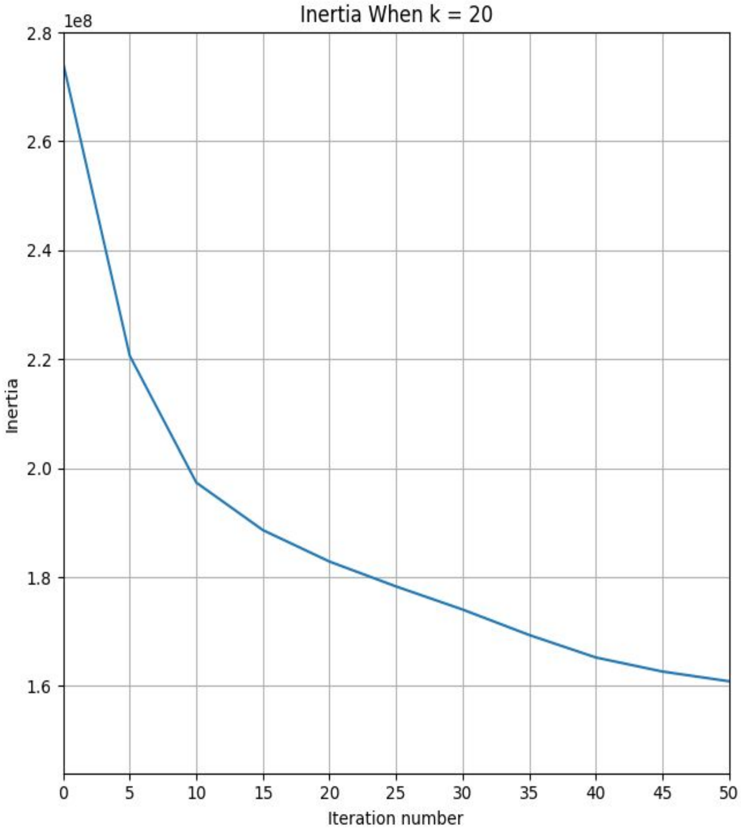
Inertia values over each iteration (to convergence) for each compressed image 3





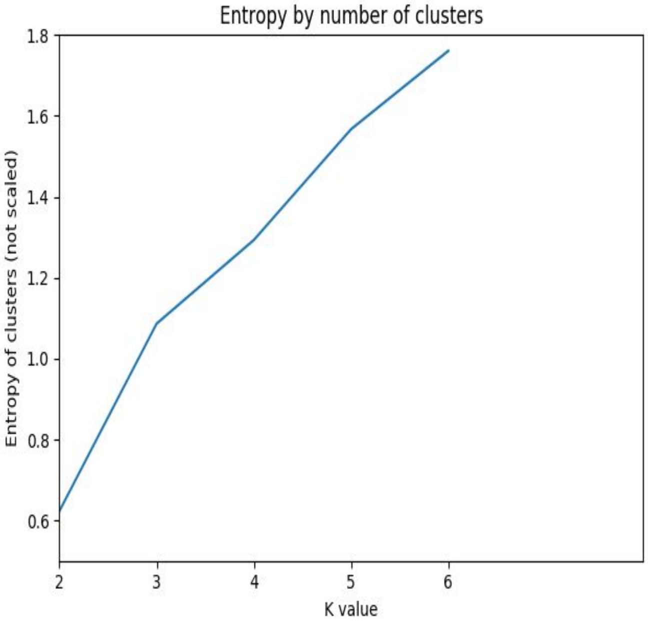

Inertia When k = 10
1e8
5.5
5.0
4.5
4.0
3.5
3.0
Inertia
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Iteration number

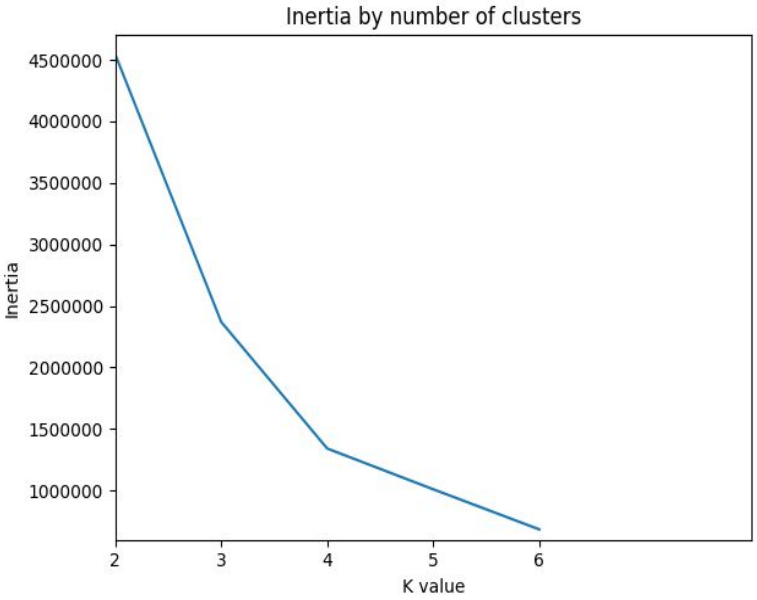

Inertia When k = 15
1e8
3.0
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
Inertia
0 5 10 15 20 25 30 35 40 45 50 55 60 65 67
Iteration number


Inertia When k = 20
1e8
2.8
2.6
2.4
2.2
2.0
1.8
1.6
Inertia
0 5 10 15 20 25 30 35 40 45 50
Iteration number

3. For each compressed image produced in the above task, plot them for each value of k, starting with the original picture. Recommend plotting these in a single row, side by side.
    a.
4. For image id 2, when k =2, explain how the color in the compressed image are not found in the original image.
    a. The reason the color in the compressed image is not in the original image is because the original image has three colors however we compress it using two clusters. What ends up happening is because the compressed image tries to take three colors and group them into two, the third color ends up being grouped into one of these clusters. This dilutes that clusters colors values and creates a new color. In our example the clustering combines yellow and red to make a new color orange.
5. Write python code to calculate out the mean silhouette coefficient for one image (you can select any EXCEPT image 2) for k = 2, 5, 10, 15, 20. Plot the results with k on the x-axis and the mean silhouette coefficient on the y-axis. Write a few (2 to 3) sentences on the quality of your clusters justifying your answer using the silhouette coefficient. You may NOT use the sklearn function, you must write your own code to calculate these values.
    a. So we tried to get this calculation, and got decently far (to the point where it does print an answer) but it is very slow and quite inaccurate. It varies in accuracy when compared to the sklearn method by a fair amount and needs more tuning. We included it in the code so you can see our attempt, but commented it out so it doesn’t impact the rest of the program.

Wine Cluster Section - k means conducted on the wine data set. Based on our results we would pick k = 4 because that is where we start to see a nice elbow curve in our inertia values. This comes at somewhat of a tradeoff as our entropy values is a bit high when k = 4.


Entropy by number of clusters
Entropy of clusters (not scaled)
K value


Inertia by number of clusters
Inertia
K value