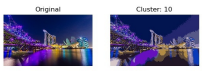
1 Exercise 1

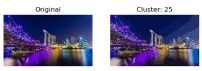
1. The elbow rule refers to a plot with x-axis **the number of clusters** and y-axis **the percentage of variance**. The percentage of variance is given by the ratio of the between-group variance to the total variance. One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data, for instance choosing k clusters, if the change in the slope between $k-1$ and k clusters is sufficiently negative and the change in the slope between k and $k+1$ clusters is close to zero. This change of the slope in k, makes the plot in k look like an elbow.

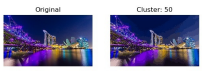
2. The kmeans++ is an algorithm for choosing the initial values for the kmeans-clustering algorithm. The intuition behind this algorithm is that spreading out the k **initial** cluster centers across the data points works out well and guarantees a solution which is comparable to the optimal solution. The algorithm proceeds as follows:
   First, the first cluster center is chosen uniformly at random from the data points. Second, for each data point, compute the distance between the data point and the nearest center that has already been chosen. Third, choose a data point at random as a new center, using a weighted probability distribution, where a data point is chosen with probability proportional to the squared distance between itself and its nearest center(computed previously). Finally, repeat the second and third steps until k cluster centers have been chosen.
   Note that the randomness in the third step, gives us a set of k clusters reasonably broadened across the data points being clustered.

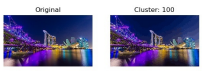
2 Exercise 2

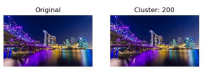
We observe that for $k = 2, 5, 10, 25$ the compressed image doesn't resemble the original image quite accurately, but for $k = 50, 100, 200$ the compressed image seems reasonably similar to the original image.











3 Exercise 3

k clusters	Reconstruction Error
2	4649.606427601365
5	2244.5552509188647
10	1194.8063190734624
25	587.0773394770773
50	340.794163091918
100	204.74784717847407
200	125.11388311817896

To compute the compression rate I used the formula :

$$\text{Compression Rate} = \frac{\text{Uncompressed size} - \text{Compressed size}}{\text{Uncompressed size}}.$$

As expected, as the number of clusters increases, the compression rate drops(the original's image distortion decreases).

k clusters	Compression Rate
2	0.958318399
5	0.874962664
10	0.833258662
25	0.791479988
50	0.749626643
100	0.707586619
200	0.665173238