Image compression using Clustering

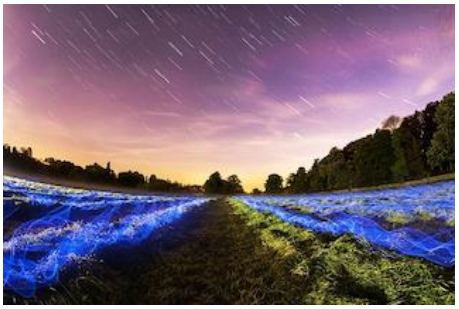
To approach this problem, an implementation of the K-means algorithm needed to be constructed. This involved the following steps:

1. Initialize random cluster centers based on the *k* selected.
2. Establish the similarity function to be used (Euclidean distance for this implementation).
3. Construct a process for assigning data points to the closest cluster (based on the similarity function).
4. Construct a process for updating the cluster centers based on the mean of each cluster.
5. Iterate through the process until there are no changes in cluster centers.

Additionally, the number of iterations and total time taken was tracked for each model. Furthermore, the images used required pre- and post-processing to view the image, run the algorithm on the array of pixels, construct the compressed image, and then view the compressed image. The result was an implementation of the K-means algorithm capable of condensing images based on the number of clusters desired.

Nature

The photo used for the K-means algorithm was tested on was the following image of a nature scene:



This picture was then compressed using *k* = 14, 16, and 18 the number of clusters. The results can be seen below:

k = *14, 16, 18*

Time Taken: 289.0 sec, Iterations: 68; 401.0 sec, Iterations: 83; 629.0 sec, Iterations: 116