E0-270: Assignment 2
K-Means Clustering on RGB Image Pixels

Gajera Pranavkumar Arvindbhai
SR No. 21073

April 09, 2023

1 Introduction

Clustering is an unsupervised machine-learning technique used to group similar data points together based on some measure of similarity or distance. K-Means clustering is a popular algorithm for this task and is widely used in various applications such as image segmentation, market segmentation, and data mining.

2 K-Means Algorithm

Algorithm 1 K-means Clustering

```
Require: $\mathbf{X}$: Dataset with $N$ instances, $K$: number of clusters
Ensure: $\mathbf{C}$: set of $K$ cluster centroids
0: Randomly select $K$ datapoints to initialize $K$ cluster centroids $\mathbf{C}^{(0)}$
0: $converged \leftarrow false$
0: while $converged = false$ do
0:    Assign each instance $\mathbf{x}_i$ to the closest centroid $c_i$:
          $$c_i = \arg \min_{x \in \{1,\dots,K\}} ||\mathbf{x}_i - \mathbf{C}_j^{(t-1)}||^2 \quad (1)$$
0:    Compute the mean of each clusters:
          $$\mathbf{C}j^{(t)} = \frac{1}{|S_j^{(t)}|} \sum_{i \in S_j^{(t)}} \mathbf{x}_i \quad (2)$$
0:    where $S_j^{(t)}$ is the set of instances assigned to cluster $j$ at iteration $t$
0:    if $\mathbf{C}^{(t-1)} = \mathbf{C}^{(t)}$ then
0:       $converged \leftarrow true$
0:    end if
0: end while=0
```

The K-Means algorithm works by iteratively assigning each data point to the nearest centroid and updating the centroids based on the mean of the assigned data points. The algorithm proceeds as follows:

The algorithm takes as input the dataset $\mathbf{X}$ and the number of clusters K, and outputs the cluster assignments $\mathbf{C}$, where each c_i denotes the cluster to which data point $\mathbf{x}_i$ belongs.

The algorithm initializes the centroids from the data points and repeats the assignment and update steps until convergence or a maximum number of iterations is reached. The assignment step involves finding the nearest centroid for each data point based on the Euclidean distance metric, while the update step involves calculating the mean of the assigned data points to update the centroid positions.

3 Experimental Setup

Implementing the K-Means algorithm first loads the sample image(512x512x3) and flattens it into a 2D array(262144x3) of pixel values normalized between 0-1.

It then initializes the k centroids randomly from the dataset and iteratively assigns each pixel to the nearest centroid, and updates the centroid positions based on the mean of the assigned pixels.

Sometimes, due to the same sampled centroids which have same RGB values , some of the centroids don't even get a single datapoint in the corresponding cluster. In that case, implementation is again to sample the new centroid from the dataset given, which was not sampled before. This process ensures the unique K centroids to cluster the given image pixels to exactly K unique clusters.

The implementation also includes a function to replace each pixel in the image with the value of its nearest centroid, resulting in a compressed information version of the original image with reduced color depth.

Implementation also has the measure of error between the original image and the image obtained with metric of mean squred error(MSE), which has less color depth compared to the original image.

4 Results

In an implementation, the K-Means algorithm was applied to a sample image of a playing child in the park as given in Figure 1 and ran K-means clustering algorithm for various values of $K = \{2,5,10,20,25,50,75,100\}$ to analyze its performance. The plot of the Mean Squared Error(MSE) as a function of the number of clusters K as calculated based on implementation of K-means clustering algoritm which is given in Table 1 to determine the optimal value for K is given in Figure 3.

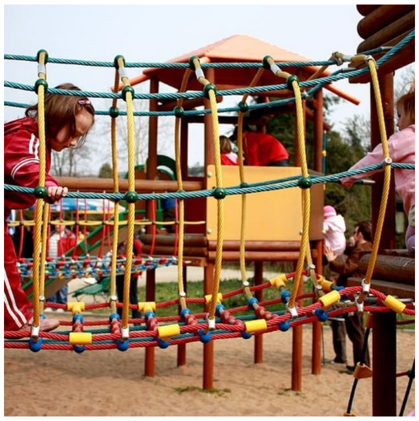

Figure 1: Original image

Figure 1 shows the original image, while Figure 2 shows the compressed version of the image with reduced color depth fo different values of $K = \{2,5,10,20,25,50,75,100\}$.

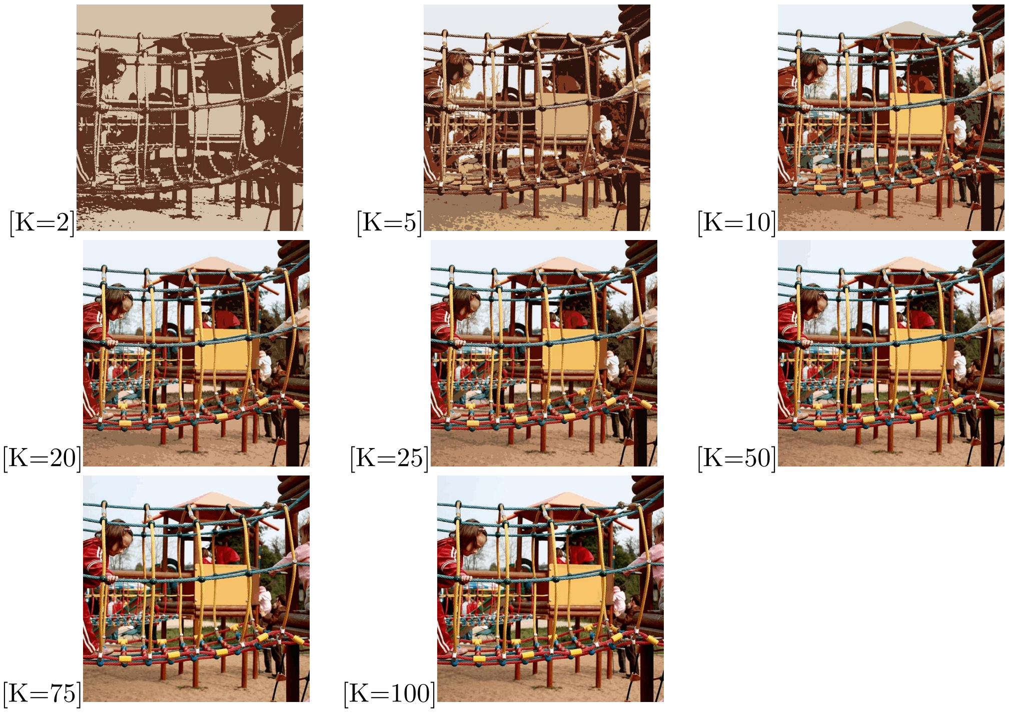

Figure 2: K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates the use for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

Table 1: Mean squared error for different values of K

K(#Clusters)	**Mean squared error(MSE)**
2	0.03206917602224996
5	0.009577374877504022
10	0.0048920065942455885
20	0.0026150632508193344
25	0.002192040119323022
50	0.001297891665852946
75	0.0009528732615442834
100	0.0007729332467604249

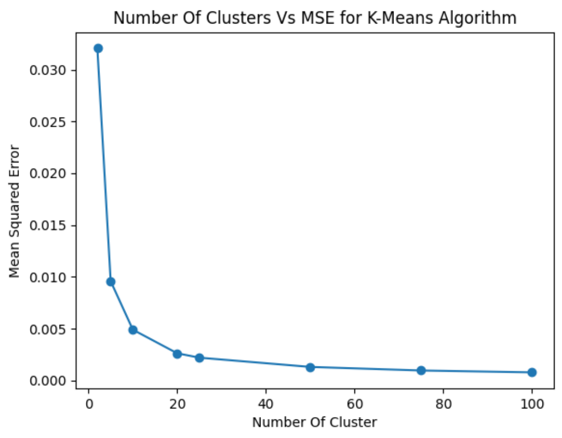

Figure 3: Plot of MSE as a function of number of clusters

Figure 3 shows the plot of MSE as a function of K. As expected, the MSE decreases as the number of clusters increases, since the centroids become more representative of the pixel values. However, the rate of decrease slows down as K increases, indicating diminishing returns in terms of compression quality.

5 Conclusion

In conclusion, Implementation of the K-Means clustering algorithm and applied it to a sample image to demonstrate its effectiveness in reducing color depth and compressing images. implementation analyzed the performance of the algorithm by plotting the Mean Squared Error as a function of the number of clusters and found that there are diminishing returns in terms of compression quality as the number of clusters increases.Here for given problem, best value of K is 20 as there is no significant changes in error afterwards for the values of K greater than 20.