

# K-means Image Compression

---

By Ashwin Mittal

In a colored image, each pixel is of size 3 bytes (RGB), where each color can have intensity values from 0 to 255. Following combinatorics, the total number of colors representing is  $256 * 256 * 256$  (equal to 16,777,216). Practically, we can visualize only a few colors in an image significantly less than the above number. So the k-means clustering algorithm takes advantage of the human eye's visual perception and uses few colors to represent the image. Colors having different values of intensity that are RGB values seem the same to the human eye. The k-means algorithm takes this advantage and clubs similar looking colors (close together in a cluster).

## Pseudocode

```
Randomly initialize k cluster centroids
Repeat
{
    for each pixel of the image:
        find the index of cluster centroid closest to it
    centroids = average of points assigned to each cluster
}
```

## Running

---

### Original Images



### Compressed Images with 16 Colors



## Explanation

---

The initial dimension of an image is  $512 * 512$  pixels. For each pixel, the image has 3-dimension representing RGB intensity values. The RGB intensity values range from 0 to 255. Since intensity value has 256 values, so the storage required to store each pixel value is  $3 * 8$  bits. Finally, the initial size of the image is  $(512 * 512 * 3 * 8)$  bits. Total number of color combination equals  $(256 * 256 * 256)$  (equal to 16,777,216). As the human eye cannot perceive so many numbers of colors at once, the idea is to club similar colors together and use fewer colors to represent the image. We will be using k-means clustering to find k number of colors representing its similar colors. These k-colors will be centroid points from the algorithm. Then we will replace each pixel values with its centroid points. The color combination formed using only k values will be significantly less compared to the total color combination. I tried  $k = 16$  and observed the output image's size.

## Output

```
lena_colorCompressed.png 112787 bytes
lena_color.png 227335 bytes
lena_grayCompressed.png 79706 bytes
lena_gray.png 264598 bytes
```

Hence, the final size of the image is reduced to a great extent from the original image.

## Conclusion

---

Finally, it is a lossy compression technique, so you have to compromise with your image quality to get the compression.