Color Quantization Using K-Means for Social Impact in Kenya's Images

Penina Wanyama
SDS6/44238/2023
MSc Data Science
School of Computing & Informatics
Penina.wanyama@students.uonbi.ac.ke

ABSTRACT

In this paper, an RGB image is processed using a color quantization algorithm based on K-Means to reduce the number of colors in the image. The color quantization algorithm selects the most representative color and reduces the ineffective color in the image as much as possible. This paper assumes that the RGB image is composed of multiple pixels. Color quantization can be realized using the K-Means algorithm to perform unsupervised clustering on these pixels with specific colors. The use of K-Means for color quantization of the images can reduce the number of colors in those images so that they can be reproduced well in lower-performance computer equipment. At the same time, color quantization reduces the size of the images and improves the efficiency of image processing such as uploading over the internet.

1. INTRODUCTION

Kenya faces challenges in healthcare, bandwidth limitations, and preserving cultural heritage due to large files from high-color images. True-color images, with thousands of colors, pose issues in display, storage, and transmission. Despite advancements in image technology, Kenya's low computer performance, low storage, and small bandwidth hinder the efficient handling of many colors. To address this, a color quantization algorithm is used for reducing he color while preserving original features.

Color quantization in Kenya can cut internet costs, optimize bandwidth, and enhance web browsing for rural mobile users. Smaller-sized, quantized images benefit healthcare imaging, support telemedicine adoption, and enable efficient storage utilization. Additionally, color quantization aids in digitizing cultural artifacts, minimizing storage needs, and promoting global access to Kenya's heritage for education and cultural appreciation.

1.1. COLOR QUANTIZATION

With the development of computer technology and the improvement of hardware performance, the collection of high-quality images is no longer an issue. Image processing technology is also widely used in various fields such as medicine, agriculture, energy exploration, etc. The color digital image in the computer is generally obtained by mixing three basic colors of red (R), green (G), and blue (B) according to a certain ratio. The colors of these images are discrete and are represented by a set of binary values between 0 and 255. The maximum number of colors stored in the computer is $2^{(8+8+8)} = 16777216$.

However, not all computers can process these high-quality images. In low-performance computers, we need to use images that are small in color but that reflect the target features in the image. Currently, it is necessary to color quantize the image with many colors. Color quantization is the merging of less important colors in an image into a relatively important color and minimizes the visual error between the original image and the quantized image. In this process, the color value of the RGB image is changed from R, G, B to R', G', B', and the formula (1) represents the color error E after color quantization.

$$E= (R-R')^2 + (G-G')^2 + (B-B')^2 \quad (1)$$

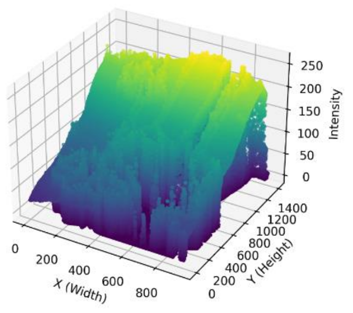


Figure 1 Three-dimensional representation of the RGB image of a coconut tree.

However, when it comes to how people see colors, it is not exactly the same as the numerical value of the colors. The paper focuses on color quantization to minimize visual errors and preserve the most characteristic colors. The goal is to make the quantized remote sensing image look as close as possible to the original image.

However, the difference in color quantization for human vision is not the same as the difference in its value. The requirements of color quantization in this paper are mainly to reduce visual error and retain the most representative color so that the quantized remote sensing image is visually as similar as possible to the original image.

2.0 Related work

In this research segment, we delve into existing literature on image compression methods, including Vector Quantization (VQ) and K-means clustering. A novel algorithm, Improved Dual-Tree Energy-Limited Block Generalized Lloyd (IDE-LBG), efficiently generates optimal Vector Quantization (VQ) Codebooks for compressing grayscale images. This method reduces the computation time and optimizes the Peak signal to Noise Ratio (PSNR).

/// demonstrating reduced computation time and excellent Peak signal-to-noise ratio (PSNR).

Another study introduces a Vector Quantization (VQ)-based image compression methodology, which was tested with various resolutions and Block Sizes. A separate study presents a K-means clustering scheme, which reduced sensor energy consumption by around 49%. The latter was evaluated through metrics like PSNR, Mean Squared Error (MSE), and Structural Similarity Index (SSIM).

Medical image compression emphasizes on size reduction while preserving quality using the Discrete Wavelet Transform - Vector Quantization (DWT-VQ) technique. Another study explores Singular Value Decomposition (SVD) for image compression. It is done by dividing the image matrix into linearly independent matrices and implementing a MATLAB-based algorithm. Image quality evaluation, including SSIM, Mean Squared Error (MSE), and Peak Signal-to-Noise Ratio (PSNR), remains crucial in compression assessment.

3.0 KMeans

Clustering is a process of dividing data into a group that shares certain characteristics following a certain pattern. KMeans clustering is a clustering technique to identify clusters of data objects as explained in **Fig. 2.** Due to the nature of clustering, it is considered unsupervised learning. Therefore, we do not need to put labels for the data as it recognizes some similar patterns between the data.

K-mean clustering is considered a centroid-based algorithm, whereby central points are chosen and the data close to each central point are grouped according to some properties. The primary process of K-mean clustering revolves around minimizing the sum of distances between points and their respective cluster centroids.

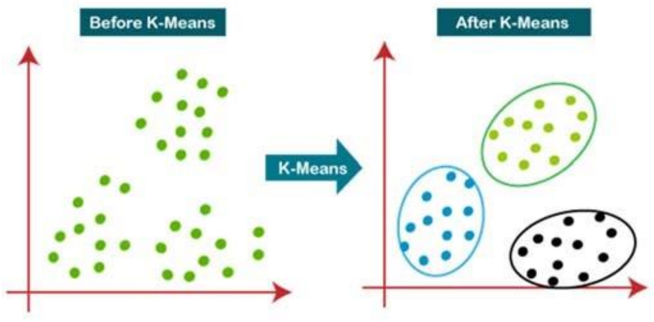
/////minimization of the sum of distances between the points and their respective cluster centroid is the main process of K-mean Clustering.

In the clustering process, the similarity between any two data points is measured by distance. Common distance measurement methods include Euclidean distance, Chebyshev distance, Manhattan distance, etc. The distance metric used in this paper is the Euclidean distance. The Euclidean distance is the linear distance between two points in the metric space. The calculation method is formula (2):

$$d=\sqrt{(x1-y1)^2+(x2-y2)^2+\dots+(xn-yn)^2}$$

$$=\sqrt{\sum_{i=1}^{n}(xi-yi)^2} \qquad (2)$$

Fig. 2



.

4.0 DATA RELEVANCE

The dataset used in the paper is a coconut tree, sourced from unsplash.com. In the original RGB image, the coconut image has 1308534 colors (1401 x 934 x 3).

The experiment environment for this article is 3.11.5, NumPy 1.24.3, Scikit-Learn 1.3.0, and Matplotlib 3.7.2. NumPy is an open-source scientific computing package that is implemented by Python. It supports multidimensional arrays and matrix operations. This experiment introduces the NumPy package for three-dimensional array operations. Matplotlib is a Python-implemented graphing/drawing package, which can display the coconut tree image and related description of the output in the visual form. Scikit-Learn is a machine learning algorithm library based on NumPy, SciPy, and matplotlib. The K-Means algorithm used in this experiment can be directly called from the Scikit-Learn library.

5.0 METHODOLOGY

Digital images are 3- dimensional arrays of pixels, and they often require to be compressed to facilitate portability and storage. K-means clustering is widely used to compress RGB images. Compression by K-mean clustering is a Lossy compression, which means compressed images cannot be restored to their original state. However, the effect on the quality of the image is directly proportional to the compression ratio, that is, if the size becomes too small compared to the original one, the quality too will drop significantly. The pseudo-code of the proposed methodology can be explained as follows:

//// The higher the compression ratio, the size decreases, but the compressed image quality will be affected. The pseudo-code of the proposed methodology can be explained as follows:

i). Pre-processing

Before applying K-Means clustering, image data (pixels) must be prepared. Images are read from the storage media, and then the data is read within (pixels with three bytes RGB color). These pixels are arranged in the form of a three-dimensional array (matrix) with several columns and rows with a color for each point as shown below.

A

$$\begin{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} & \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} & \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \\ \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} & \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix} & \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \end{pmatrix}$$

The image matrix A is first flattened to a 2D array by multiplying the rows and the columns of the matrix to enable K-Means to be effective in clustering.

If we assume *h* in A is the rows of the matrix, ***w*** in A is the columns of the matrix, and c represents the 3 color channels, we can flatten the 3D array to a 2D using the below formula and the resulting array will be shown below. 2_D_A

$$\begin{bmatrix}\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\end{bmatrix}$$

In NumPy, it can be implemented using the. **reshape*()*** function

After K-Means clustering processing, the dimension is again transformed into 2D and then to 3D to form the image like the original one.

ii) Apply the K-Means technique

1. Select K data points from a data set containing **n** data points as the initial cluster center **C**$_1$, where the data points are the pixels of the image. We will set our K=[16,32,64,128,256]
2. Calculate the Euclidean distance *d* (L2 norm) between each data point and the cluster centers separately with the idea of the magnitude of a vector X at points x_1,x_2 ..x_n:

$$||x||_2 = \sqrt{(x_1)^2 + (x_2)^2 + \ldots (x_n)^2}$$

3. Suppose the data point coordinates are (**x**$_1$, **y**$_1$, **z**$_1$) and the cluster centre coordinates are (**x**$_2$, **y**$_2$, **z**$_2$). The calculation method is shown in formula (4), and the data point with the smallest distance from the cluster center is classified as the same category as the cluster center.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_{21})^2} \qquad (4)$$

4. Calculate the mean according to the existing data points of each category, reselect the cluster centres of each category according to formula (5)

$$C_i = \frac{1}{n_j} \Sigma_{x_j \epsilon C_i} x_j \qquad (5)$$

This is repeated until the cluster center C_i doesn't change.

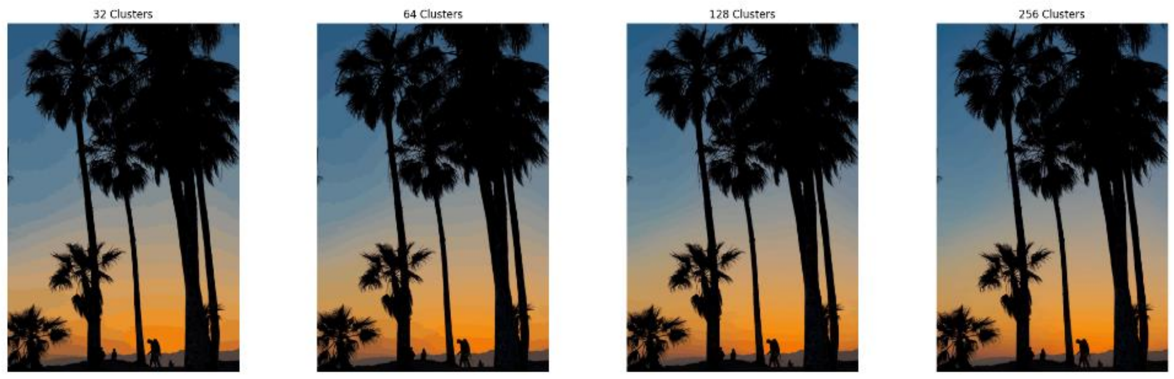
In a nutshell, K-Means clustering aims to organize row data into K groups, $S= \{S_1, S_2, ...S_k\}$ based on numerical model. The goal is to minimize the specific formula (formula 6) to achieve classification where 'k' represents the number of desired groups ad it should be less than or equal to total data point 'n'.

//// In an nutshell, the purpose of K-Means clustering is to divide the raw data into K classes $S = \{S_1, S_2, ...S_k\}$ given the number of classification groups k is less than n ($k \leq n$), on the numerical model, which is to find the minimum value of the formula (6):

$$argmin \sum_{i=0}^{k} \sum_{x_j \epsilon S_i} ||x_j - \mu_i||^2$$

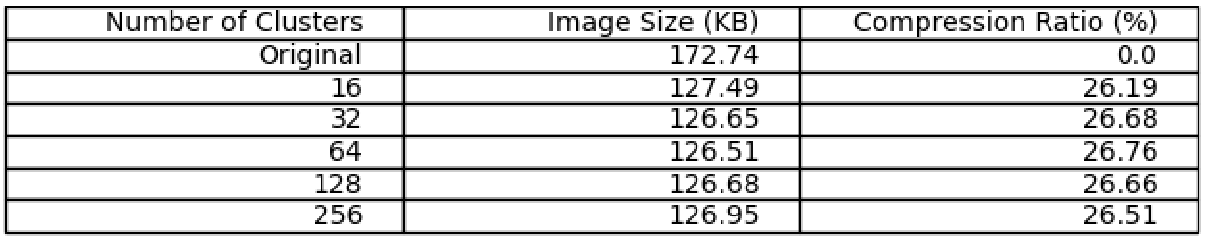
Where μ_i represents the average of S_i

5. Use the "pyplot" module of the matplotlib package to visualize the output, as shown in Figure 3 below.



6. Derive the quantized map and compare the size of the original image with the quantized image, see **Table 1**.

Image Size and Compression Ratio Comparison



Number of Clusters	Image Size (KB)	Compression Ratio (%)
Original	172.74	0.0
16	127.49	26.19
32	126.65	26.68
64	126.51	26.76
128	126.68	26.66
256	126.95	26.51

5.0 Results Analysis

The results (Figure 3) of the 16,32,64,128 and 256 color clusters in the image color quantization were obtained. We note that the size of the reproduced image is generally smaller than the original image after the compression process. An increase in the number of clusters increases the size of the reproduced image. The first line is the original image.

From the two experiments, it can be found that the quantized image of the second row contains more color features than the third row. Therefore, the color quantization algorithm based on K-

Means has a small visual error in the color quantization of images, which can better reflect the color characteristics of remote sensing images.

As can be seen from the table (Table 1) and image visualizations above, a major advantage of color quantization is that the size of the RGB image is reduced, which greatly saves the storage space of the RGB image in the computer. When the computer processes the image, the RGB image with less memory can improve the processing speed and improve the research efficiency.

6.0. CONCLUSION

K-Means-based algorithms are applied to RGB coconut tree. In the experiment, the number of cluster centres was set to 16,32,64,128 and 256, respectively, and the experimental results were obtained. A comprehensive analysis of the results of the experiment yielded the following conclusions:

I. The color quantization algorithm based on K-Means has a better effect on the color quantization of images.
II. Colour quantization of RGB images can reduce image size.

Therefore, the result of using the color quantization algorithm based on K-Means has a smaller visual error and consumes less memory space. Color quantization not only enables RGB images to be reproduced in more low-performance devices, improves image usability, but also saves computer storage space. It also improves image processing and transmission over the internet efficiently.

REFERENCES

1. Yang Jinfeng, Chen Qing, Han Xiaori et al. Application of Digital Image Processing Technology in Leaf Area Measurement of Vegetables [J]. Transactions of the Chinese Society of Agricultural Engineering, 2002, 18(4): 155-158 (in Chinese). DOI = 10.3321/j.issn:10026819.2002.04.038
2. Cheng Guojian, Ma Wei, Wei Xinshan, et al. Rock Fabric Recognition Based on Image Processing and Neural Network [J]. Journal of Xi'an Shiyou University: Natural Science Edition, 2013, 28(5): 105-110 (in Chinese).DOI = 10.3969/j.issn.1673-064X.2013.05.021
3. REN Zhibin, ZHAI Yongxin, YANG Yinghui, YANG Huaijiang.Color quantization of color
   images in uniform color space[J].Optics and Precision Engineering,2002(04):340-345 (in Chinese). DOI = 10.3321/j.issn:1004-924X.2002.04.004
4. Ozturk C , Hancer E , Karaboga D . Color Image Quantization: A Short Review and an Application with Artificial Bee Colony Algorithm [M]. IOS Press, 2014. DOI = 10.15388/Informatica.2014.25
5. Long Y , Gong Y , Xiao Z , et al. Accurate Object Localization in Remote sensing images Based
   on Convolutional Neural Networks[J]. IEEE Transactions on Geoscience & Remote Sensing, 2017, 55(5):2486-2498. DOI = 10.1109/TGRS.2016.2645610