Majestic

Majestic team presents

PR project

Image compression

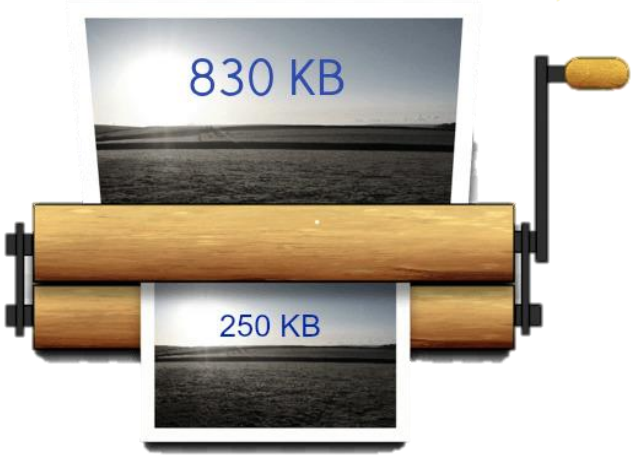
Table of contents

01

Project Description

Project Description

Project name	Image Compression
The algorithm used to pattern recognition “Compression algorithm”	K-means clustering algorithm
The intended pattern in the project	Color pixel in digital image
Input	Digital image
Output	The compressed version of the input image.
The purpose of project	To reduce the size of the image while maintaining an acceptable level of visual quality for storage or transmission..

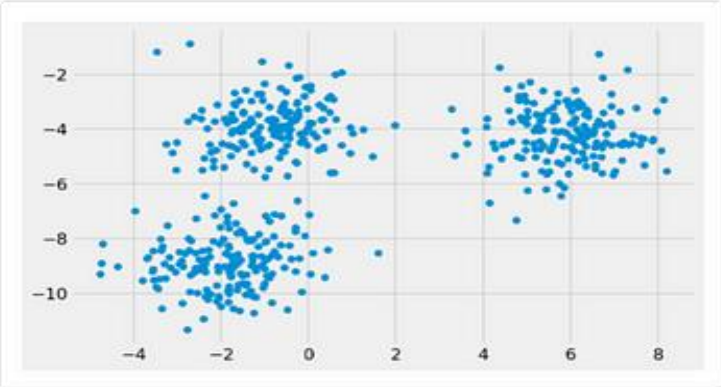


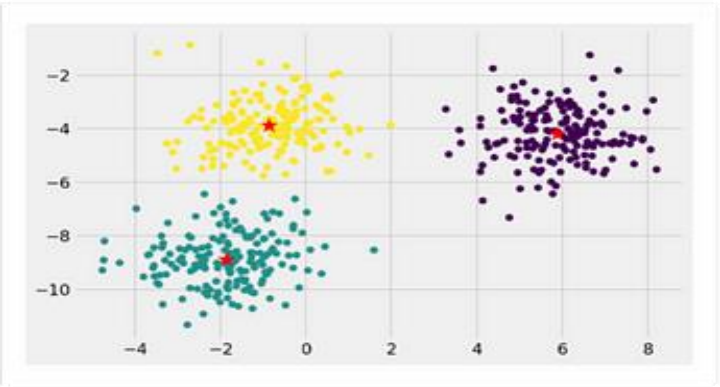
What does image compression mean?

Image compression is a type of data compression applied to digital images without degrading the quality of the image to an unacceptable level.

What is K-Means Clustering?

K-Means algorithm is a centroid based clustering technique. This technique cluster the dataset into k different clusters. Each cluster in the k-means clustering algorithm is represented by its centroid point.





Left Image: Plot of the dataset,

Right Image: Plot of the result of 3-means clustering.

Why do we use k-means algorithms to compress images?

Simplicity

It is simple and easy to implement, making it accessible to a wide range of users and applications.

Speed and Scalability

It is a fast and scalable algorithm used for processing large datasets in real-time applications, and its versatility has made it popular in computer vision, machine learning, and data compression.

Flexibility

It is a versatile algorithm that can be used for both lossy and lossless compression of images, depending on the number of clusters used, making it adaptable to different applications and requirements.

Good performance:

It achieves high compression ratios and acceptable visual quality, particularly for images with large uniform areas. It's useful for compressing maps, diagrams, and charts.

02

project mechanism

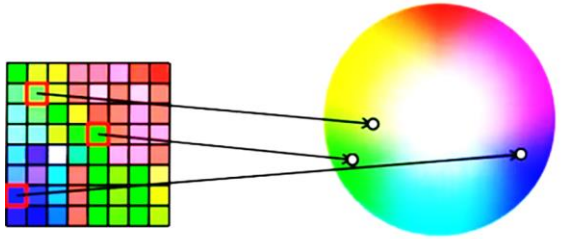
The steps of the k-means algorithm for image compression

Step 1: Convert input RGB image to Lab color space using rgb2lab from skimage.color.
Step 2: Reshape Lab image to a 2D array of pixels (L, a, b values).
Step 3: Randomly initialize k centroids from flattened image without replacement.
Step 4: Repeat for specified iterations:

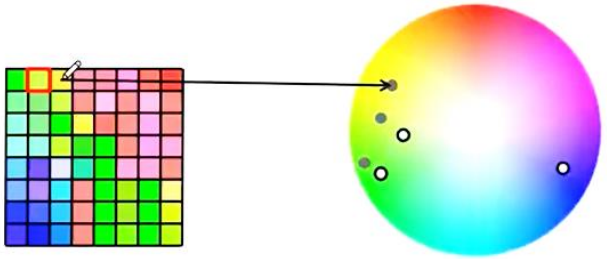
- **a.** Calculate Euclidean distance between each pixel and each centroid.
- **b.** Assign each pixel to nearest centroid based on minimum distance.
- **c.** Update each centroid to be the mean of the assigned pixels.

Step 5: Convert compressed Lab image to RGB color space using lab2rgb from skimage.color and reshape to original shape.
Step 6: Clip pixel values to range [0, 1] and multiply by 255 to obtain 8-bit unsigned integer representation.
Step 7: Return the compressed image.

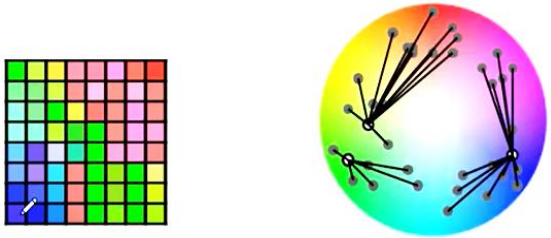
Explain clustering the image pixel in figures



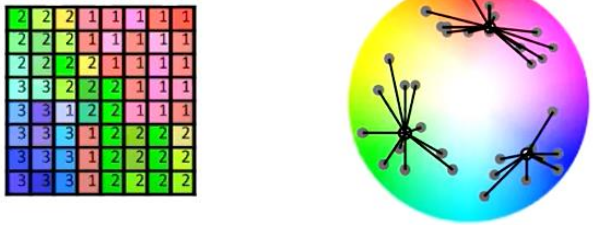
Step 3



Step 4.a

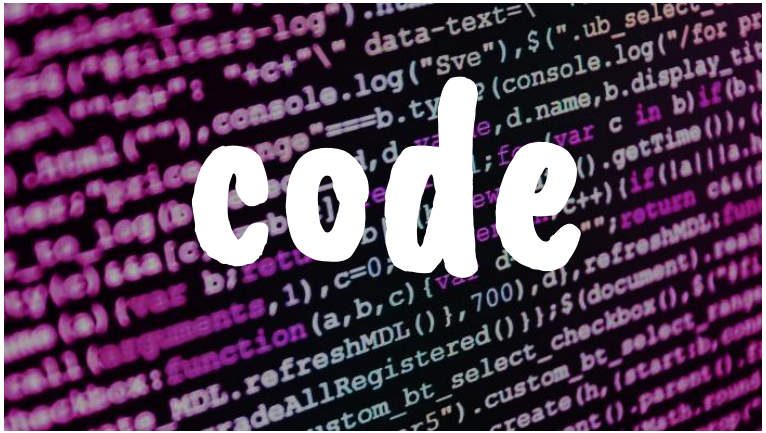


Step 4.b



Step 4.c

03



The libraries imported in the code

1 

NumPy used for numerical computations, array manipulation, and mathematical operations.

2 

PyQt is used to create graphical user interfaces (GUIs) and desktop applications with a native look.

3 

skimage.color provides color-related functions and used to convert the image between RGB and Lab color spaces..

4 

matplotlib.pyplot is used for creating visualizations, including displaying images and plotting the WCSS curve in this code.

5 

OS Module

os provides a way of using operating system-dependent functionality and used to get the size of the images files.

The component functions of the code

k_means_lab

that performs the k-means clustering algorithm on an input image.

plot_WCSS_curve

that plots the WCSS curve for a range of k values and shows the optimal k value.

calculate_wcss_value

that calculates the Within-Cluster Sum of Squares (WCSS) for a given set of centroids and their corresponding assigned cluster datapoints

comparison_images

that compares the original and compressed images and displays their sizes and color information.

elbow_method

that used to determine the optimal number of clusters (k) for compressing an image.

main

that compresses an input image using the k-means algorithm with a given k value or the optimal k value (determined by the elbow_method function).

The Application by PyQt

Graphical user interface (GUI) created with PyQt5, a Python module that provides bindings for the Qt application framework. The GUI is for an image compression project and allows the user to select an image, perform K-means clustering on the image, display the within-cluster sum of squares (WCSS) curve, and switch between light and dark modes.

MyWindow class

The MyWindow class inherits from the QWidget class and defines the main window of the GUI.

The __init__ method

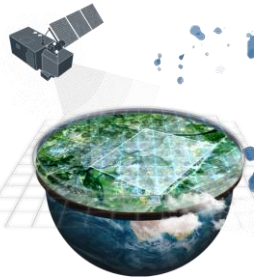
- initializes the window by setting various attributes such as the window title, size, icon, and layout. It also creates various widgets such as labels, buttons, and spin boxes, and connects them to their respective methods.

04

Project application area

Image compression using k-means clustering has many applications in real life scenarios .

where large amounts of image data need to be stored or transmitted over limited bandwidth networks. **Here are some examples of its applications:**



- **Digital cameras:** Compressing images to save storage space.
- **Online image sharing:** Compressing images to make them easier and faster to upload and download.
- **Medical imaging:** Compresses medical images like X-rays and CT scans to reduce storage space and network bandwidth required.
- **Satellite imaging:** Compressing images to reduce the amount of data that needs to be transmitted back to Earth.
- **Video streaming:** Compressing individual frames of videos to reduce file size and improve streaming performance.
- **Autonomous vehicles:** Compressing visual data to reduce the amount of data that needs to be processed in real time.

Prepared By :

Ahmed Diaa ELShahat

Raafat Mohamed El-menayar

Wesam Ahmed El-Sharawy

Ibrahim Mohamed El-Morsi

Ahmed Abdel Raouf Abdel Shafi

Under the supervision of :

Dr.Mona Elbedwehy