Application of the K-Means Algorithm in Image Compression

Hongxia Mao

School of Computer and Software, Jincheng College of Sichuan University, Chengdu 611731, Sichuan, China

Abstract: ***In the Internet era, image information is widely used across all industries. The volume of image data is enormous, so effective transmission and storage require image compression. The K-Means algorithm is one of the most commonly used clustering algorithms. This paper applies the clustering approach to image compression and uses Python to implement K-Means clustering for image compression. Experimental results show that K-Means clustering can indeed compress images.***

Keywords: K-Means; Clustering; Image compression.

1. INTRODUCTION

Clustering analysis is an important research area in data mining. It groups data objects into several classes or clusters so that objects within the same cluster are similar, while objects in different clusters differ greatly. Image information is widely used across all industries; the greater the information content of an image, the larger the space it occupies. Therefore, appropriate compression methods are required for image transmission and storage. Image compression refers to reducing the amount of data needed to represent an image by eliminating redundancy while maintaining a certain quality, thereby facilitating storage and transmission.

This paper applies the clustering approach to image compression, using the K-Means clustering algorithm to compress images. In addition, this also helps to reduce the after-sales costs and reputation losses caused by defective products for enterprises. In network optimization, [1] proposed Log2Learn, an intelligent log analysis framework for real-time system performance enhancement. Educational applications are explored by [2], whose AI-powered data analysis enables early detection of learning difficulties through pattern recognition. Computer vision techniques are advanced by [3] through their YOLOv8-based algorithm for vehicle detection in traffic imagery. The evaluation of AI empathy is addressed by [4], who introduced EmotionQueen as a benchmark for assessing large language models' emotional intelligence. In healthcare, [5] systematically reviewed deep learning approaches for personalized ECG diagnostics, highlighting solutions for inter-patient variability. Dermatological applications are demonstrated by [6], whose IoT-based system employs active learning for skin cancer detection with optimized hyperparameters. For human-AI interaction, [7] developed InVis, an interactive neural visualization system facilitating user-centric data interpretation. Reliability engineering is advanced by [8] through RAID, an automated detection framework for large-scale advertisement systems. Developer tooling innovations are presented by [9] via InfraMLForge, which accelerates LLM development and deployment. Finally, [10] contributed to generative AI applications with GenPlayAds, enabling procedural generation of interactive 3D advertisements.

2. K-MEANS CLUSTERING ALGORITHM

Clustering is a very important application area in data mining. Clustering, based on the principle of similarity, groups samples with high similarity into the same cluster and places samples with high dissimilarity into different clusters. The K-Means algorithm is one of the most commonly used clustering algorithms. Also known as the K-means algorithm, it uses distance as the similarity measure between samples; the smaller the distance, the higher the similarity, and the more likely they are to belong to the same cluster. The algorithm takes a parameter k and partitions the sample points into k clusters; samples within the same cluster have high similarity, while samples in different clusters have low similarity. The idea is to cluster around k sample points in space, assigning the closest samples to each cluster. Through iterative updates of the cluster centers, the best clustering result is achieved.

The algorithm proceeds as follows:

(1) Randomly select k samples from the dataset of n samples as the initial cluster centers;

(2) Calculate the distance from each sample in the dataset to the k cluster centers and assign the sample to the cluster whose center is closest;

(3) Recalculate the cluster center for each category;

(4) Repeat steps 2 and 3 until the cluster centers no longer change.

Overall, K-means the clustering idea of the algorithm is simple and easy to implement, and the clustering effect is acceptable; it is a simple, efficient, and widely used clustering method.

3. APPLICATION OF THE K-MEANS ALGORITHM IN IMAGE COMPRESSION

3.1 Principle of Image Compression Using the K-Means Algorithm

When an image is displayed on a computer screen, it occupies memory space. The formula for calculating the memory occupied by an image is: image height * image width * memory size per pixel. The number of bytes per pixel directly affects the memory occupied by the image. Storing images in different color modes requires different amounts of memory, as shown in Table 1:

Table 1: Comparison of Memory Occupied per Pixel for Different Image Types

图像类型	每像素多少字节	可表示的颜色数量
1 比特数据图 (Line art)	每像素1/8 字节	$2^1=2$
8 比特灰度 (Grayscale)	每像素 1 字节	2^8
16 比特灰度 (Grayscale)	每像素 2 字节	2^{16}
24 比特 RGB	每像素 3 字节,这是 图片中最常用的,如 TIF 格式。	2^{24}
32 比特 印刷色彩 模式 (CMYK)	每像素 4 字节	2^{32}
48 比特 RGB	每像素 6 字节	2^{48}

The basic principle of image compression is to replace similar colors with a single color, reducing the number of colors described. Consequently, each pixel can be described with fewer bytes, and the memory occupied by the image decreases. The basic principle of using the K-Means clustering algorithm for image compression is to classify each pixel in an image and replace the values of pixels belonging to the same class with the cluster center value of that class, thereby reducing the memory space occupied by the image.

3.2 Experimental Process

When importing an image into the program, the image data needs to be preprocessed. According to the image resolution, the data points are flattened, treating each pixel as a 3-dimensional sample. The pixel values of an image are converted into a n by 3 data matrix, where n = is height * width.

Use the KMeans clustering algorithm, set the k value, cluster all color values in the image, and find the cluster center corresponding to each 3-dimensional pixel point.

The number of colors in the compressed image is the number of clusters

K=64
kmeans = KMeans (n_ clusters =k, random state=0)

Train the model

kmeans.fit(pixel_sample)

Find the cluster center corresponding to each 3-dimensional pixel point

cluster_assignments = kmeans.predict(pixel_sample)

Traverse every pixel in the image, find the cluster-center pixel value corresponding to each pixel value, and replace

the pixel's value with that cluster-center value.

Traverse each pixel and find the pixel value of its cluster center

```
pixel_count =0
for i in range(height):
for j in range(width):
```

Get the index of the cluster center for this pixel

```
cluster_idx = cluster_assignments[pixel_count]
```

Get the pixel value at the cluster-center index

```
cluster_value = cluster_centers[cluster_idx]
```

Replace the pixel's value

```
compressed_img[i][j] = cluster_value
pixel_count +=1
```

Run the code; Figure 1 shows the image before compression, and Figure 2 shows the image after compression using the K-Means algorithm k=64.

Figure 1: Effect before compression

Figure 2: Effect after compression

3.3 Experimental Results

According to the runtime results, the original image is 92KB (bird.TIF), whereas the compressed image is only 45K (compressed bird), achieving the goal of image compression.

- compressed_dog 2019/8/9 2:05 JPG image file 45KB

(C) 2017/10/23 11:11 JPG image file 92 KB

4. CONCLUSION

From the experimental results, the K-Means clustering method can indeed compress images; the smaller the value of k, the higher the compression ratio, but the fewer colors remain in the compressed image, resulting in significant loss of pixel color information and greater deviation from the original. The K-Means algorithm is simple and easy

to implement, yet it requires the user to specify the number of clusters in advance (the value of k), and the clustering results are sensitive to the initial choice of cluster centers, making it prone to local optima. In practice, to obtain better results, K-Means is usually run multiple times with different initial cluster centers. After all samples have been assigned, when recalculating the centers of the k clusters, the cluster center for continuous data should be the mean of that cluster.

REFERENCES

[1] Tu, T. (2025). Log2Learn: Intelligent Log Analysis for Real-Time Network Optimization.
[2] Wang, Chun, Jianke Zou, and Ziyang Xie. "AI-Powered Educational Data Analysis for Early Identification of Learning Difficulties." The 31st International scientific and practical conference “Methodological aspects of education: achievements and prospects”(August 06–09, 2024) Rotterdam, Netherlands. International Science Group. 2024. 252 p.. 2024.
[3] Wang, Hao, Zhengyu Li, and Jianwei Li. "Road car image target detection and recognition based on YOLOv8 deep learning algorithm." unpublished. Available from: http://dx. doi. org/10.54254/2755-2721/69/20241489 (2024).
[4] Chen, Yuyan, et al. "Emotionqueen: A benchmark for evaluating empathy of large language models." arXiv preprint arXiv:2409.13359 (2024).
[5] Ding, Cheng, et al. "Advances in deep learning for personalized ECG diagnostics: A systematic review addressing inter-patient variability and generalization constraints." Biosensors and Bioelectronics (2024): 117073.
[6] Yang, Jing, et al. "IoT-Driven Skin Cancer Detection: Active Learning and Hyperparameter Optimization for Enhanced Accuracy." IEEE Journal of Biomedical and Health Informatics (2025).
[7] Xie, Minhui, and Shujian Chen. "InVis: Interactive Neural Visualization System for Human-Centered Data Interpretation." Authorea Preprints (2025).
[8] Zhu, Bingxin. "RAID: Reliability Automation through Intelligent Detection in Large-Scale Ad Systems." (2025).
[9] Zhang, Yuhan. "InfraMLForge: Developer Tooling for Rapid LLM Development and Scalable Deployment." (2025).
[10] Hu, Xiao. "GenPlayAds: Procedural Playable 3D Ad Creation via Generative Model." (2025).