DOI: 10.21917/ijsc.2020.0307

A COMPARATIVE ANALYSIS OF IMAGE COMPRESSION TECHNIQUES: K MEANS CLUSTERING AND SINGULAR VALUE DECOMPOSITION

R. Gomathi and R. Aparna
Department of Computer Applications, Madras University, India

Abstract
The global drive to digitize almost all the existing processes has mandated the conversion of all concerned analog data into their respective digital formats. One such crucial data that is being digitized on a priority in today's world is image. An image is a type of data which is composed of picture elements called pixels. It can be represented as a matrix for the manipulating process. The storage of a vast database of image files occupies a huge memory space in the disk. To overcome this hassle, image files can be compressed and saved. This image compression process is aimed at reducing the data size in terms of bytes and enable the efficient storage and transmission of image files. Image compression can be achieved through several algorithms. In this paper, we discuss two such algorithms, namely k means clustering and singular value decomposition. K Means Clustering technique helps in minimizing the colour components of the image. Singular Value Decomposition technique can be carried out by low rank approximation of the image matrix. This research work is performed using the Python platform and subsequently the efficiency of both the methods is compared. The comparative analysis of the simulation results are further compared with the existing methods to show the competence of different methodologies. Thus, this work strives to be of learned assistance to the concerned aspirants in choosing the best algorithm for their applications.

Keywords:
Image Compression, K-Means Clustering, Singular Value Decomposition

1. INTRODUCTION

The digital streamlining of even the conventional day to day routines has caused the usage of multimedia to surge tremendously on a daily basis. The frequently used multimedia format in this regard is Image. Every day we collect and store many images for various reasons and purposes. This activity leads to the enormous storage of image files which occupy almost all the memory space of the computer disk. A good solution for this problem is using compression techniques [1] to reduce the size of the image. There are various compression techniques, and we can choose the appropriate technique based on the type of image and the desired quality of the image output after compression.

Image Compression techniques can be classified into two types: lossless compression, and lossy compression. Lossless compression [2] retains all the original data and there will not be any degradation in the image quality even after compression. If needed, it can be easily decompressed into its original form. It is mostly used to compress the medical and business images in order to maintain the image quality. Lossy compression [3] removes the redundant data permanently. Thus, the compressed file cannot be changed into its original form.

Image compression is achieved by eliminating unnecessary data. The changes are made by removing redundant data that are futile and lost during image processing. These redundant data are not visible to the human eye in most cases (psychovisual redundancy). The basic redundancies which are easily identified are coding redundancy, interpixel redundancy and psychovisual redundancy.

Coding redundancy is related to representing information in the form of code. The coded images have more codes than the necessary limit required to represent information, and hence contain coding redundancy. Interpixel redundancy occurs when there is relation between the neighbouring pixels. The value of any pixel in the image can be found with the help of its neighbouring pixels due to the correlation between them. Psychovisual redundancy takes place when the human eye cannot identify certain degraded or unimportant pixels in the image.

The two compression techniques that are discussed in this paper are lossless compression, and lossy compression techniques. They are *K* Means Clustering based image compression and Singular Value Decomposition (SVD) based image compression.

The composition of the remaining parts of this paper are as follows: section 2 furnishes the literature survey. Section 3 and Section 4 deal with the elaboration on K Means Clustering and Singular Value Decomposition respectively. Section 5 reveals the experimental results. Section 6 evaluates the simulated results, and section 7 presents the conclusion of this paper.

2. LITERATURE SURVEY

Image signal comprises of a huge dataset and hence it is hard to proceed with other image processing techniques. So, it is essential to compress the image dataset without losing any important data. Zhang and Xiaofei [1] have discussed various compression techniques by learning to minimize the total error mechanisms. RGB and Gray scale component on MPQ-BTC is given in [2]. Gunjan Mathur et al. [3] have elaborated the importance of various lossy compression techniques. K Means clustering has various benefits over other techniques. This unsupervised method is discussed in [4] – [7]. Singular Value Decomposition technique has been proved to be one of the most frequently used compression methods. It is evidently stated in [8]. Mounika et al. [9] supports a lot in achieving good PSNR values using SVD. Thus, a strong background work has been carried to proceed with this paper.

3. IMAGE COMPRESSION USING K MEANS CLUSTERING

K Means Clustering algorithm [4] is a type of unsupervised learning, which divides the data points into *k* predefined clusters based on the minimal distance from the centroid. We can represent the image as an array of [R, G, B] values, where height and width of the image are taken as row and column values of the array. The algorithm is initialized by selecting random set of values, around which many other values are grouped [5]. Then all the values in the cluster are replaced based on the centroid which is near it. Iterative process of these steps will reduce the number of colours used in the image. As a result, the image will be compressed successfully.

At first, randomly select $K(\mu_k)$ centroids from the pixel data points $\{x^{(1)}, x^{(2)}, x^{(3)},\ldots, x(N)\}$, where x represents the pixel values and N refers to the number of pixels in the image. Calculate the distance $\sum\|x^i - \mu_k\|^2$ between each data point $(x(i))$ where $i = 1,2,3,\ldots,N$ and centroid(μ_k). Assign the pixel data points to the cluster which has minimum distance from the centroid. Update the centroids by recalculating it. Mean of the data points obtained from each cluster will be the new centroids for each cluster. Follow these two steps until there is no change in centroids.

To solve the problem, the approach followed by K Means is Expectation Maximization. The purpose of E-step [6] is to assign data points to the nearest centroid. The M-step is to calculate the centroid of each cluster. It can be represented mathematically as,

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}\left\|x^i - \mu_k\right\|^2 \tag{1}$$

If pixel data point x_i belongs to cluster k, then w_{ik}=1; otherwise w_{ik} = 0. μ_k is the centroid of x_i cluster.

The preceding function Eq.(1) is the reduced equation of two parts [7]. At first, we differentiate J with respect to w_{ik} and keep μ_k as fixed which updates cluster assignments (E-step). Then we differentiate J with respect to μ_k and keep w_{ik} fixed and recalculate the centroids after the cluster assignments from previous step (M-step). Therefore E-step is:

$$\frac{\partial J}{\partial r} = \sum_{i=1}^{N}\sum_{k=1}^{K}\left\|x^i - \mu_k\right\|^2 \Rightarrow w_{ik} = \begin{cases} 1 & if\ k = \arg\min\left\|x^i - \mu_k\right\|^2 \\ 0 & otherwise \end{cases} \tag{2}$$

In other words, assign the pixel data points x_i to the nearest cluster which is obtained by calculating the sum of squared distance from the cluster's centroid.

The *M*-step is:

$$\begin{aligned} \frac{\partial J}{\partial \mu_k} &= 2\sum_{i=1}^{N} w_{ik}\left(x^i - \mu_k\right) = 0 \\ \Rightarrow \mu_k &= \frac{\sum_{i=1}^{N} w_{ik}x^i = 0}{\sum_{i=1}^{N} w_{ik} = 0} \end{aligned} \tag{3}$$

3.1 ALGORITHM

Step 1: Read the image (input) as an array.

Step 2: Fix the value for K and number of iterations.

Step 3: Initialise the algorithm by selecting K number of centroids.

Step 4: Compute the distance and assign the pixel data points to the nearest cluster to which the centroid belongs.

Step 5: Update the centroids by computing the mean of data points in the cluster.

Step 6: Repeat step 4 and step 5 till there is convergence in centroids or until the number of iterations are completed.

Step 7: Construct the image using the array which has undergone the clustering algorithm.

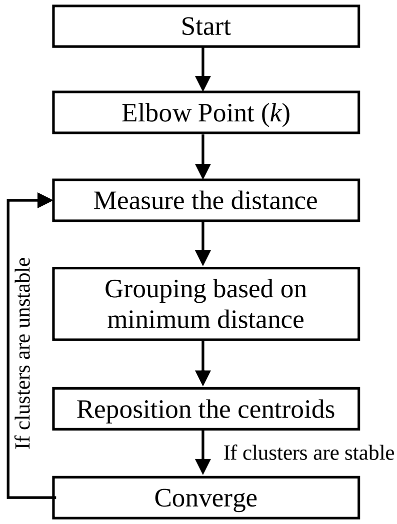


Fig.1. Flowchart of K Means Clustering Algorithm

3.2 IMAGE COMPRESSION USING K MEANS - KEY CONSIDERATIONS

- The K must be chosen before starting the algorithm.
- Always K must be smaller than number of image pixels, i.e. (K<N).
- Distance Formula used here is Euclidean distance. Euclidean distance is the absolute value of the difference between data point and centroid.
- Numbers of iterations should be mentioned as priority.

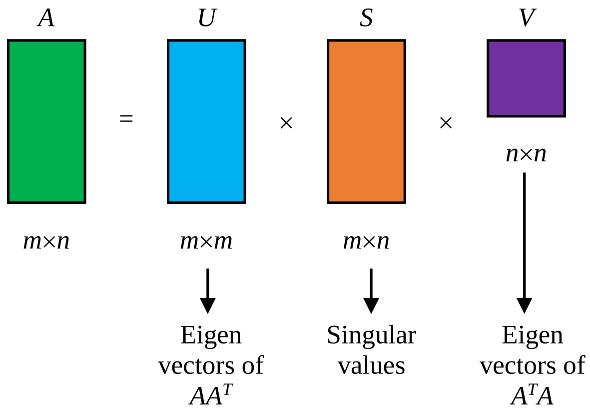


Fig.2. Representation of Singular Value Decomposition

4. SINGULAR VALUE DECOMPOSITION BASED IMAGE COMPRESSION

Singular value decomposition algorithm is one of the matrix factorisation technique [8]. The aim of SVD is to find the best approximation of original pixel data points that are of huge dimensions, with fewer dimensions. Any $m\times n$ image can be represented as $m\times n$ matrix. Pixel values of the image are the elements of the $m\times n$ matrix. An image matrix A can be decomposed into $A = USV^T$ [9].

The Fig.1 states that A is a real $m\times n$ matrix, U (orthogonal) is a $m\times m$ matrix containing orthonormal Eigen vectors of AA^T, S is an $m\times n$ matrix consisting of singular values which are the positive square root of non-zero Eigen values of A^TA and V(orthogonal) is a $n\times n$ matrix containing the orthonormal Eigen vectors of A^TA.

The Singular Value Decomposition algorithm removes redundant data by performing low rank approximation. The most crucial data are stored in certain singular values. So, we can eliminate some of the singular values for bringing about better compression.

4.1 STEPS TO CALCULATE SVD OF A MATRIX

Step 1: At first, compute AA^T and A^TA.

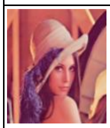
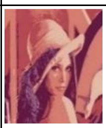
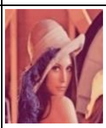
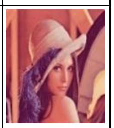
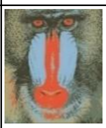
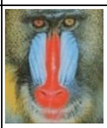
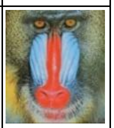
Step 2: Use AA^T and A^TA to compute Eigen values and Eigen vectors to make up the column of U and V.

Step 3: To form the columns of U and V, divide each Eigen vector by its magnitude.

Step 4: Calculate the square root of non-zero Eigen values and arrange them in the diagonal matrix S in descending order: $\sigma_1\geq\sigma_2\geq\sigma_3,\ldots,\geq\sigma_r\geq 0$, where $r = \min(m, n)$ [10].

Step 5: Reconstruct the matrix by multiplying U, S and V.

Table.1. Compressed Images under K Means Clustering Algorithm

Original Image		K=5	K=16	K=30	K=60
	Lenna				
	Baboon				
	Pepper				

4.2 PROPERTIES OF SVD

- The SVD decomposes matrix A into USV^T with K real singular values.
- The singular values of a rectangular matrix A are equal to the positive square roots of non-zero Eigen values $\lambda_1, \lambda_2,\ldots, \lambda_m$ of the matrix A^TA [11].
- The rank of the matrix A is equal to the number of its positive Eigen values, rank(A) = r, $r\leq m\leq n$.
- The singular values 'σ' are unique but the matrices U and V are not unique.
- The Euclidean norm of A is equal to the largest singular value, $||A||^2= \sigma_1$

4.3 ALGORITHM

Step 1: Read the image as an array.

Step 2: Fix the value for K and Number of iterations.

Step 3: Isolate the red, green and blue channels of the image.

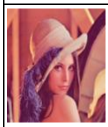
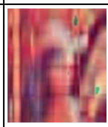
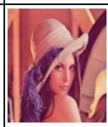
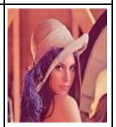
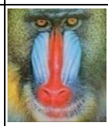
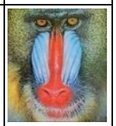
Step 4: Perform SVD on each channel.

Step 5: Merge all the channels together to reconstruct the image.

5. EXPERIMENTAL RESULTS

In this research work, image compression is performed using the K-means clustering algorithm and Singular Value Decomposition algorithm. The results obtained are tabulated as follows:

Table.2. Compressed Images under Singular Value Decomposition Algorithm

Original Image		K=5	K=16	K=30	K=60
	Lenna				
	Baboon				
	Pepper				

In Table.2, K refers to number of colours. We can differentiate each image based on colour. The Table.1 shows that when K is small, the number of colours used in the image will also be small [12]. Similarly, it is shown that the number of colours increases if the value of K increases. The goal of this compression algorithm is to reduce the size by minimizing the number of colour components in the image. If we want to maintain the image quality, K can be increased. When K is large, compression ratio becomes large; when K is small, compression ratio becomes small. If there is large number of pixels, number of iterations must be increased to get better results in compression.

The redundant data in the image are discarded which later leads to compression. The Table.2 shows that Smaller the K value, lesser the image quality. By increasing the K value, we can get the desired output. To reduce the file size, we need to set the smaller K value. To maintain both size and quality we should balance and find the optimum K value [13].

6. EVALUATION

PSNR: The term Peak Signal to Noise Ratio is an expression which calculates the ratio between maximum possible value of a signal and the power of distorting noise that affects the quality of its representation (in decibels) [14]. The value obtained from PSNR is used as quality measurement between the original image and a compressed image. Higher the PSNR value, better the image quality. PSNR can be calculated by,

$$PSNR = 10\log_{10}\left(\frac{(L-1)^2}{MSE}\right) \tag{3}$$

SSIM: The Structural Similarity Index quantifies image quality degradation that are caused by processing such as data compression or data transmission. It measures the perceptual structural differences between two similar images. Unlike PSNR, SSIM is based on visible structures in the image. SSIM is computed as,

$$SSIM_{(x,y)} = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \quad (4)$$

Compression Ratio: Compression ratio is defined as the ratio between original image size and compressed image size. Higher the compression rate, better the output results. It tells how well an image is compressed when compared with its original image. It is calculated as,

$$CR = \text{original image size/compressed image size} \quad (5)$$

Compressed image has same dimensions as the original image. Only the size differs since it undergoes compression. When the *k* value decreases, the size of the compressed image also decreases. We can visualise it in the above graphs. When *K* =60 in *K* Means Clustering and *K* = 150 and 230 in SVD, the quality of image is so good and the colours which are lost due to *K* means clustering based image compression are not visible to human eye as they are psycho-visual redundancy. When *K*=5 in both algorithm, there are too many degradations in the image quality and it is visible to human eye.

Table.3. Evaluation results of Compressed Images

Algorithm	Evaluation Parameters			
	Image	**PSNR**	**SSIM**	**CR**
K Means Clustering (*K* = 60)	Lenna	31.77	0.8402	11.78
	Baboon	25.28	0.8007	8.18
	Peppers	31.67	0.8596	8.36
SVD (*K* = 60)	Lenna	29.07	0.8581	12.58
	Baboon	21.68	0.5924	9.79
	Peppers	21.36	0.8400	7.44
Existing Method 1 [1]	Lenna	33.00	-	23.7
	Baboon	25.2	-	11.4
	Peppers	31.0	-	22.6
Existing Method 2 [2]	Lenna	24. 1209	-	-
	Baboon	23.9565	-	-
	Peppers	24.1531	-	-

The Table.3 demonstrates the quality, structural differences, and compression ratio between the two algorithms. We can visualize the rate of distortion and structural similarity between original image and compressed image by PSNR and SSIM. The Table.3 gives a clear analysis of various compression techniques which helps the researchers to choose the best method for their research. As the image signal is highly vulnerable, it is essential to secure the compressed image before using it in the emerging trends like IoT [15], deep learning, big data analytics, etc.

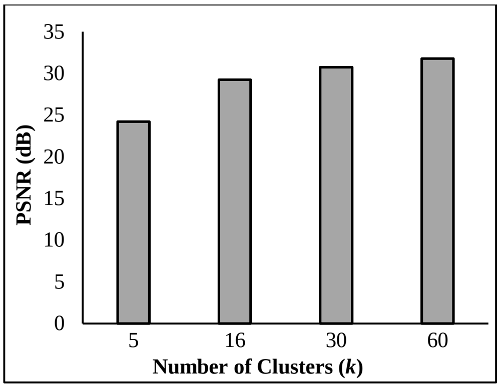


Fig.3. Plot of *K* value vs. PSNR in K-means

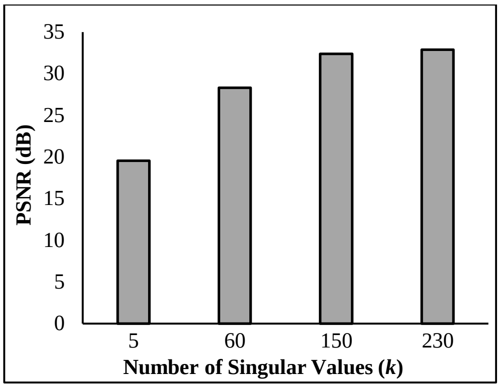


Fig.4. Plot of Singular Values vs. PSNR in SVD

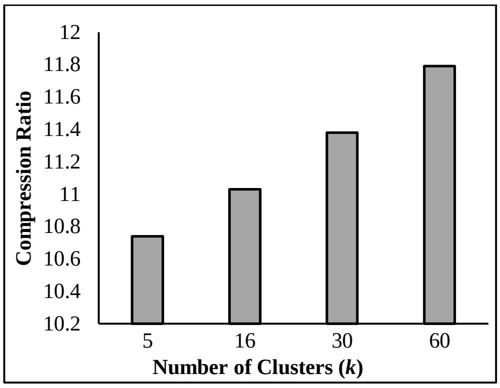


Fig.5. Plot of *K* value vs. Compression Ratio in K-means

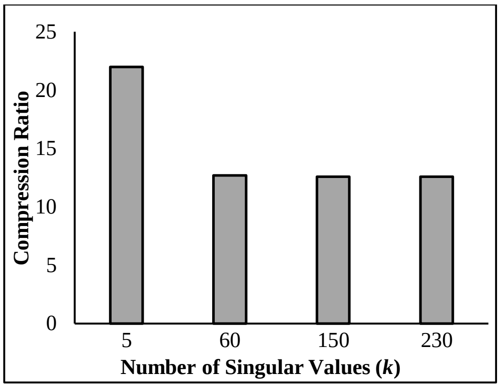


Fig.6. Plot of Singular value vs. Compression Ratio in SVD

The Fig.2 and Fig.3 disclose the relation between *K* value and PSNR. We can clearly see that, if the *K* value increases, the PSNR also increases slowly. The compressed image with higher K value and PSNR resembles the original image without any degradation in quality. The Fig.5 and Fig.6 shows the correlation between *K* value and compression ratio. In Fig.5, compression ratio increases with increase in *K* and in Fig.6, compression ratio decreases with increase in *K* value which is inversely proportional to each other.

7. CONCLUSION

Image compression is an essential requisite for today's digital lifestyle due to the proliferation of large image files across all media. The techniques proposed and enumerated by the preceding parts of this paper certainly accomplish the task of compressing the size of image files in an efficient manner. As evident from the discussions delineated above, we can observe that, if the *K* value is smaller, the compression rate in both the algorithms are higher. By analysing and comparing these two algorithms we can see that the output is better in the *K* Means Clustering Algorithm with lower *K* value. These techniques can be used for both personal and business purposes. Comparative analysis of both the two considered algorithms with other existing techniques shall help to identify the efficient method.

REFERENCES

[1] Chiyuan Zhang and Xiaofei He, "Image Compression by Learning to Minimize the Total Error", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 23, No. 4, pp. 565-576, 2013.

[2] K. Somasundaram and P. Sumitra, "RGB and Gray Scale Component On MPQ-BTC In Image Compression", *International Journal on Computer Science and Engineering*, Vol. 3. No. 4, pp. 1-13, 2011.

[3] Gunjan Mathur, Rohit Mathur and Mridul Kumar Mathur, "A Comparative Study of Various Lossy Image Compression Techniques", *International Journal of Engineering Research and Technology*, Vol. 2, No. 3, pp. 1-9, 2018.

[4] A. Harika, V. Revanth Narayana, C.H. Sagar Kumar and Suresh Kurumalla, "Reduction of Redundant Pixels of an Image using Efficient K Means Clustering Algorithm", *International Journal of Engineering Sciences and Research Technology*, Vol. 8, No. 3, pp. 104-110, 2019.

[5] R.D. Sivakumar and K. Ruba Soundar, "E-Learning Image Compression using K-Means-BTC in Public Cloud", *International Journal of Current Engineering and Scientific Research*, Vol. 5, No. 5, pp. 111-119, 2018.

[6] Xing Wan, "Application of K-Means Algorithm in Image Compression", *IOP Conference Series: Materials Science and Engineering*, Vol. 563, pp. 1-5, 2019.

[7] N. Venkateswaran and Y.V. Ramana Rao, "K- Means Clustering Based Image Compression in Wavelet Domain", *Information Technology Journal*, Vol. 6, pp. 148-153, 2007.

[8] Poonam Dhumal and S.S. Deshmukh, "Survey on Comparative Analysis of Various Image Compression Algorithms with Singular Value Decomposition", *International Journal of Computer Applications*, Vol. 133, No. 6, pp. 1-12, 2016.

[9] K. Mounika, D. Sri Navya Lakshmi, K. Alekya and M.R.N. Tagore, "SVD Based Image Compression", *International Journal of Engineering Research and General Science*, Vol. 3, No. 2, pp. 1-5, 2015.

[10] Ranjeet Kumar, Utpreksh Patbhaje and A. Kumar, "An Efficient Technique for Image Compression and Quality Retrieval using Matrix Completion", *Journal of King Saud University-Computer and Information Sciences*, pp. 1-9, 2019.

[11] K.M. Aishwarya, Rachana Ramesh, Preeti. M. Sobarad and Vipula Singh, "Lossy Image Compression using SVD Coding Algorithm", *Proceedings of International Conference on Wireless Communications, Signal Processing and Networking*, pp. 1-7, 2016.

[12] Jeongyeup Paek and JeongGil Ko, "K-Means Clustering-Based Data Compression Scheme for Wireless Imaging Sensor Networks", *IEEE Systems Journal*, Vol. 11, No. 4, pp. 1-12, 2017.

[13] Sana Shafik Desai and M.S. Chavan, "Comparative Analysis of Singular Value Decomposition (SVD) and Wavelet Difference Reduction (WDR) based Image Compression", *International Journal of Engineering Research and Technology*, Vol. 10, No. 1, pp. 1-14, 2017.

[14] Yusra A.Y. Al-Najjar and Der Chen Soong, "Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI", *International Journal of Scientific and Engineering Research*, Vol. 3, No. 8, pp. 1-5, 2012.

[15] R. Nandhini and S.R. Aparna, "Study of Security Issues in Internet of Things", *International Journal of Research and Analytical Reviews*, Vol. 5, No. 3, pp. 1-12, 2018.