Scientific Journal for Publishing in Health Research and Technology

Volume 2, Issue 1, 2026, Pages: 110-125

Journal homepage: https://sjphrt.com.ly/index.php/sjphrt/en/index

Digital Image Compression Based on Singular Value Decomposition Algorithm

Zarrug A. Ganiya [1]*, Alghannai Aghnaiya [2]

[1,2] Department of Communications, College of Electronic Technology, Bani Waleed, Libya

ضغط الصور الرقمية باستخدام خوارزمية تحليل القيم المفردة

الزروق عبدالقادر غنيّة،[1*]، الغناي غنيّة [2]

[2،1] قسم الاتصالات، كلية التقنية الإلكترونية، بني وليد، ليبيا

*Corresponding author: Zarrugzz@gmail.com

Received: October 14, 2025	Accepted: December 20, 2025	Published: January 24, 2026

Abstract:

Image compression represents a fundamental application of data compression techniques within the field of digital image processing. As digital images contain vast amounts of information, there is a critical need for efficien t methods to store and transmit large data volumes. This research explores the Singular Value Decomposition (*SVD*) algorithm as a robust mathematical framework for achieving image compression by leveraging low-rank matrix approximations. The primary objective is to implement the *SVD* algorithm and evaluate its performance based on specific metrics, namely Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (*MSE*). The study provides a detailed investigation into the trade-off between the compression ratio and the resulting image quality. Methodologically, the *SVD* process factorizes an image matrix A into three distinct components: U, S, and V^T. By retaining only, the first r singular values—which contain the maximum signal energy—the algorithm can effectively reconstruct an approximation of the original image using significantly less storage space. Experimental simulations were conducted using MATLAB on various test images, including grayscale (such as Lena and Baboon) and full-color RGB images. Results demonstrate that image clarity improves as more singular values are reintroduced. For a 256 * 256 grayscale image, a close resemblance to the original was achieved using only 90 singular values, yielding a PSNR of 37.42 dB. In color images, the process involves decomposing the red, green, and blue saturation matrices separately before recombination. The findings confirm that while higher rank values increase image fidelity, they simultaneously reduce the compression ratio. Ultimately, SVD is shown to be a stable and effective numerical method for splitting data into signal and noise subspaces, providing a practical solution for modern digital communication requirements.

Keywords: Singular Value Decomposition, Image Processing, Image Compression, SVD, PSNR, MSE.

الملخص

يعد ضغط الصور تطبيقاً أساسياً لتقنيات ضغط البيانات في مجال معالجة الصور الرقمية .ونظراً لأن الصور الرقمية تحتوي على كميات هائلة من المعلومات، تبرز حاجة ماسة لطرق فعالة لتخزين ونقل أحجام البيانات الكبيرة .يستكشف هذا البحث خوارزمية تحليل القيم المفردة (SVD) كإطار رياضي قوي لتحقيق ضغط الصور من خلال استغلال تقريب المصفوفات ذات الرتب المنخفضة. الهدف الأساسي هو تنفيذ خوارزمية SVD وتقييم أدائها بناءً على مقاييس محددة، وهي ذروة نسبة الإشارة إلى الضجيج (PSNR) ومتوسط مربع الخطأ.(MSE) تقدم الدراسة تحليلاً دقيقاً للمفاضلة بين نسبة الضغط وجودة الصورة الناتجة .من الناحية المنهجية، تقوم عملية SVD بتحليل مصفوفة الصورة Aإلى ثلاث مكونات متميزة : Uو Sو V^T. ومن خلال الاحتفاظ بأول r من القيم المفردة فقط - والتي تحتوي على أقصى طاقة للإشارة - يمكن للخوارزمية إعادة بناء تقريب للصورة الأصلية باستخدام مساحة تخزين أقل بكثير .أُجريت محاكاة تجريبية باستخدام برنامج MATLAB على صور اختبار مختلفة، بما في ذلك الصور ذات التدرج الرمادي (مثل لينا وبابون) والصور الملونة بالكامل بنظام.RGB أظهرت النتائج أن وضوح الصورة يتحسن مع زيادة عدد القيم المفردة المستخدمة .بالنسبة لصورة رمادية بحجم 256 * 256، تم تحقيق تشابه كبير مع الصورة الأصلية باستخدام 90 قيمة مفردة فقط، مما أعطى نسبة PSNR بلغت 37.42 ديسيبل .وفي الصور الملونة، تتضمن العملية تحليل مصفوفات تشبع الألوان الأحمر والأخضر والأزرق بشكل منفصل قبل إعادة تجميعها. وتؤكد النتائج أنه بينما تزيد قيم الرتب العالية من دقة الصورة، فإنها تقلل في الوقت نفسه من نسبة الضغط .وفي الختام، أثبتت SVD أنها طريقة عددية مستقرة وفعالة لفصل البيانات إلى مساحات فرعية للإشارة والضجيج، مما يوفر حلاً عملياً لمتطلبات الاتصالات الرقمية الحديثة.

الكلمات المفتاحية: تحليل القيم المفردة، معالجة الصور، ضغط الصور، SVD، PSNR، MSE.

Introduction

One of the main challenges associated with digital images is the presence of excessive redundant and irrelevant information stored along with each captured picture. Many images suffer from issues such as blurriness, fading, and noise, which degrade quality and increase storage requirements. Technically, this unwanted information is referred to as noise, defined as data that is irrelevant or meaningless in the context of the image.

Image compression is a technique designed to efficiently encode digital images by reducing the number of bits required for their representation. With the advancement of digital technology, there has been a growing demand for high-efficiency compression techniques, especially since many of the traditional algorithms are computationally intensive and block-based in execution [3].

Visual information plays a critical role in modern communication systems. Applications such as high-definition television (HDTV), video conferencing, medical imaging, wireless video transmission, virtual reality, video telephony, and video servers all rely heavily on efficient image compression. As the number of users and the resolution of content increases, the amount of visual data to be transmitted or stored grows significantly, placing a substantial burden on channel bandwidth and storage capacity. Despite improvements in transmission and storage technologies, the cost of supporting high-capacity systems rises sharply with increased demand.

Compression techniques are essential in addressing these limitations, as they allow for significant reduction in data rates while preserving the perceptual quality of the image or video. Effective compression exploits redundancies inherent in image data. These include:

1. **Spatial redundancy**: arises from correlations among neighboring pixels.
2. **Spectral redundancy**: results from correlations between different color channels or bands.

3. **Temporal redundancy**: occurs in video sequences due to similarities between consecutive frames.

Temporal redundancy, in particular, can be addressed using interframe coding techniques such as Motion Compensated Predictive Coding (MCPC) [2].

Over the years, numerous image and video compression algorithms have been developed, some of which have become widely adopted international standards. Examples include JPEG, MPEG, H.261, and H.263, each offering different trade-offs between compression ratio, computational complexity, and output quality [1].

Singular Value Decomposition (SVD) is widely recognized as an optimal matrix factorization technique in the least squares sense, concentrating the maximum signal energy into the fewest possible coefficients [1], [2]. It is a stable and powerful numerical method used to decompose a matrix into a set of linearly independent components, each representing a portion of the system's energy. In the field of numerical analysis, SVD is commonly employed for matrix diagonalization [3], [4].

Due to its numerous advantages, SVD has become a highly attractive algebraic tool in image processing applications. Among its most notable features are its energy compaction capability, which is particularly useful in image compression [5], [6], and its ability to separate data into two orthogonal subspaces: the signal (information) subspace and the noise subspace [6], [7], [8]. This property has been successfully applied not only in compression but also in noise filtering and digital watermarking applications [9], [6].

The main objective to implement image compression algorithm based on SVD algorithm and evaluate the performance of this algorithm in terms of PSNR and MSE.

SVD

SVD is an approach of advanced linear algebra [14]. It is based on the packing the maximum energy of a signal into a lesser number of coefficients. It is an effective method to split a matrix into linearly independent constituents where each constituent has its own contribution in terms of energy. The uses of SVD are diverse ranging from areas such an image processing, latent semantic analysis, approximation of the pseudo inverse of a matrix, least square minimization of a matrix, efficient medical imaging, topographical analysis, watermarking schemes and many other areas. In the case of image compression, SVD offers its advantage in the form of its sensitivity to local adaptations in the statistics of an image. The core mathematical foundations of SVD can be summarized as factorizing a matrix A into three components U, known as the matrix of rows, S called the diagonal matrix or the singular values of A and V is called the matrix of columns. These factors of the matrix satisfy the relation $A = U*S*V^T$.

For a given Matrix A of size m×n the output of SVD has the following components.

U: a matrix of dimension $m \times m$.

S: the diagonal matrix of dimension $m \times n$,

V: a matrix of dimension $n \times n$ and V^T represents the transpose of the matrix V.

The orthogonal matrices U and V are not the same — since A need not be square, U and V need not even have the same dimensions. The columns of U are left singular vectors of A. The columns of V (that is, the rows of V^T) are right singular vectors of A. The entries of S are the singular values of A. Thus, with each singular vector (left or right) there is an associated singular value. The "first" or "top" singular vector refers to one associated with the largest singular value, and so on [15,16]. See Fig.1.

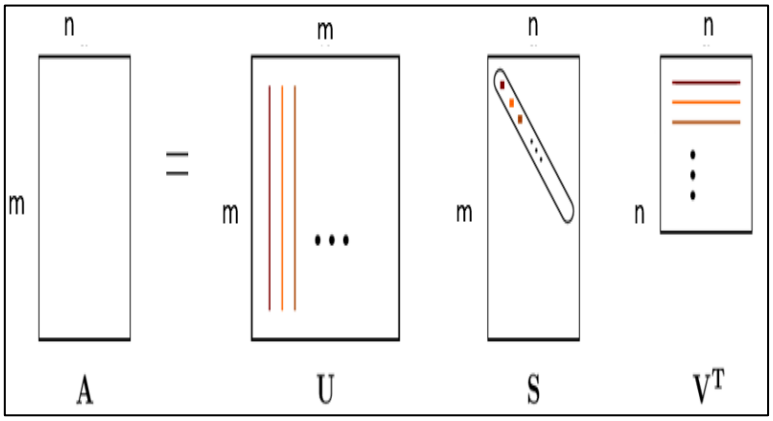


fig1: The singular value decomposition (SVD).

SVD in Image Compression:

The objective of image compression is to represent an image with lesser amount of data than what an image is composed of and the ability to reconstruct the image from its smaller representation [16]. This improves the storage efficiency of an image and also greatly reduces the amount of data that is required to transmit the image across computers. However, the image formed from its compressed image by an image processing algorithm may or may not be able to recreate the exact copy of original image. A compression technique can be lossy or lossless based on the quality of image it restores. A lossless compression scheme can reconstruct the exact copy of an image whereas a lossy scheme can recreate the image with some data loss, depending on the compression technique used. We have used SVD as lossy image compression scheme. The other methods for image compression are discrete wavelet transform, discrete cosine transform, Karhunen-Lohve transform, and combinations of these. The reason why we have used singular value decomposition because it is basic, simple and works almost for all kind of matrix and it is well suited for image compression [20,21]. As images are stored in the form of matrices in the computer memory, it is imperative to think an image as a matrix. Depending on amount of type of image, colored or grayscale, the space required to store an image depends on the dimension of the image. A grayscale image has the space requirement of m×n where m and n denote the height and the width of the image whereas a colored image has the space requirement of $m \times n \times 3$, as there are 3 matrices of $m \times n$ each representing the colors red, green and blue commonly known as the RGB image. From the properties of SVD it follows that a matrix A can be represented in the form of its SVD components as a sum of rank 1 matrices of the form:

$$A = U_1 * S_1 * V_1^T + U_2 * S_2 * V_2^T \ldots\ldots\ldots\ldots U_n * S_n * V_n^T$$

In the above relation, it is worth mentioning that the value of $S_1 > S_2 > S_3 > \ldots S_n$. The above relation also implies that, the contribution of the first component of the sum would be highest while the contribution of the last component would be lowest. Thus, it follows that if we consider only the first r members of the above summation, we can still get a considerable approximation of A. This is the property used for SVD based image compression. The relation for the compression of an image considering the first r singular values can be show to be:

$$A_r = U_1 * S_1 * V_1^T + U_2 * S_2 * V_2^T \ldots\ldots\ldots\ldots U_n * S_r * V_r^T$$

Here A_r represents the approximation of the image based on the first r singular values of the singular matrix S. Thus, instead of storing the matrix A of size $m \times n$, we can store the matrices U_{mxr},V_{nxr} and the singular vector S_r and reconstruct the image as: $A_r = U_{mxr} * S_r * V^T_{nxr}$. Thus, it

leads to a reduction in the amount of space needed to store the image and the space complexity of the compressed image would be given by $A_r = r\ (m + n + 1)$ [22,23].
Depending on the value r of the rank of SVD selected, we can get a compression ratio that would be defined as:

$$C_r = \frac{m*n}{r(m+n+1)}$$

Mean square error (MSE):

$$MSE = \frac{1}{mn}\sum\left(O_{ij} - R_{ij}\right)^2$$

where O represents the original image and R represents the reconstructed image of dimension m*n.

Peak signal to noise ratio:

(PSNR) = 10 log (255^2/MSE)

Mathematical Example of SVD.

Now we will get into the math and theory computing of the SVD. We will go through an example to solve the equation $A = U\Sigma V^T$

$$\text{Given} \qquad A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

The first thing we need to find in this computation is the matrix AA^T. The superscript T stands for "transpose" which to put nicely, you flip the matrix on its side, row one becoming column one. In order to find U, we have to start with AA^T.

The transpose of A is $$A^T = \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 17 & 8 \\ 8 & 17 \end{bmatrix}$$

The characteristic polynomial is

$$\det (AA^T - \lambda I) = \lambda^2 - 34\lambda + 225 = (\lambda - 25)(\lambda - 9),$$

$$\text{For } AA^T \qquad \lambda = 25, 9$$

The singular values are

$$\sigma_1 = \sqrt{25} = 5 \text{ and } \sigma_2 = \sqrt{9} = 3.$$

Now we find the right singular vectors (the columns of V) by finding an orthonormal set of eigenvectors of $A^T A$. It is also possible to proceed by finding the left singular vectors (columns

of U) instead. The eigenvalues of A^TA are 25, 9, and 0, and since A^TA is symmetric we know that the eigenvectors will be orthogonal.

$$A^TA = \begin{bmatrix}3 & 2\\ 2 & 3\\ 2 & -2\end{bmatrix}\begin{bmatrix}3 & 2 & 2\\ 2 & 3 & -2\end{bmatrix} = \begin{bmatrix}13 & 12 & 2\\ 12 & 13 & -2\\ 2 & -2 & 8\end{bmatrix}$$

$$\begin{bmatrix}13 & 12 & 2\\ 12 & 13 & -2\\ 2 & -2 & 8\end{bmatrix} - \lambda\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$= \begin{bmatrix}13-\lambda & 12 & 2\\ 12 & 13-\lambda & -2\\ 2 & -2 & 8-\lambda\end{bmatrix}$$

For A^TA λ = 25, 9, 0.

For λ = 25, we have

$$A^TA - 25\,\mathrm{I} = \begin{bmatrix}-12 & 12 & 2\\ 12 & -12 & -2\\ 2 & -2 & -17\end{bmatrix}$$

$$\begin{bmatrix}-12 & 12 & 2\\ 12 & -12 & -2\\ 2 & -2 & -17\end{bmatrix}\begin{bmatrix}v1\\ \\ \end{bmatrix} = \begin{bmatrix}0\\ 0\\ 0\end{bmatrix}$$

Which row-reduces to $\begin{bmatrix}1 & -1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}$. A unit-length vector in the kernel of that matrix

$$\vec{v}_1 = \begin{bmatrix}1\\ 1\\ 0\end{bmatrix} \qquad normalized \rightarrow \qquad \vec{v}_1 = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\\ 0\end{bmatrix}$$

For λ = 9 we have

$AA^T - 9\,\mathrm{I} = \begin{bmatrix}4 & 12 & 2\\ 12 & 4 & -2\\ 2 & -2 & -1\end{bmatrix}$. Which row-reduces to $\begin{bmatrix}1 & 0 & -\frac{1}{4}\\ 0 & 1 & \frac{1}{4}\\ 0 & 0 & 0\end{bmatrix}$.

$\vec{v}_2 = \begin{bmatrix}1\\ -1\\ 4\end{bmatrix}$ $normalized \rightarrow$ A unit-length vector in the kernel is $\upsilon_2 = \begin{bmatrix}1/\sqrt{18}\\ -1/\sqrt{18}\\ 4/\sqrt{18}\end{bmatrix}$

For the last eigenvector, we have: $normalized \rightarrow$ $\upsilon_3 = \begin{bmatrix}2/3\\ -2/3\\ -1/3\end{bmatrix}$

$$\vec{V} = \begin{bmatrix}1/\sqrt{2} & 1/\sqrt{2} & 2/3\\ 1/\sqrt{18} & -1/\sqrt{18} & -2/3\\ 2/3 & -2/3 & -1/3\end{bmatrix}$$

So, at this point we know that $A = U\Sigma V^T = U\begin{bmatrix}5 & 0 & 0\\ 0 & 3 & 0\end{bmatrix}\begin{bmatrix}1/\sqrt{2} & 1/\sqrt{2} & 0\\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18}\\ 2/3 & -2/3 & -1/3\end{bmatrix}$

Finally, we can compute U by the formula $\sigma u_i = A\upsilon_i$, or $u_i = \frac{1}{\sigma} A\upsilon_i$.

This gives $U = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$. So, in its full glory the SVD is:

$$A = U\Sigma V^T = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{bmatrix}$$

Results and Discussion

we evaluated the performance of SVD algorithms that were implemented in MATLAB. The studied algorithm is applied on several test images with different singular values (ranks). The results of the meticulous simulation for the images are presented and compared in terms of (MSE, PSNR, and CR).

Grayscale image compression using (SVD) with different ranks

The process of SVD can be used for compress images to conserve storage space by removing the singular values that contribute the least to the information contained in the image matrix. For this simulation, we choose grayscale image, as shown in Fig.2. We design MATLAB code to load an image, and isolate the corresponding saturation matrix, and then modify the matrix based on its singular values. As an example, we use a high-contrast grayscale image. We consider the individual saturation levels of each pixel in the original image as the numerical entries in a matrix. We compute the SVD of that matrix and remove the singular values (from smallest to largest), converting the modified matrices (with removed values) back into a series of images. This process of decomposition can reduce the image storage size without losing the quality needed to fully represent the image. In Fig.2. we can see that as more singular values are included in the image matrix; the clarity of the image improves. The original image has approximately 256 singular values, but we were able to see a close resemblance to the original image using only 90 singular values. The amount of storage space is not as significant in our example here as it would be in practice, because of our emphasis on image clarity. Our current process is to compress while still retaining the original number of pixels in order to show the details of the loss of image quality. In practice, we would see a more significant change in storage of an image if we allowed the overall image size (the number of pixels) to reduce as we removed the small singular values. In Fig.4, Table.1, we can see the amount of error is increased when the values of singular values (rank) are decreased. We observe the positive concavity of the error curve, which indicates that as the error decreases, the rate of change of the error loss (MSE) also decreases. This means that the rate of change of the error loss is less significant as more singular values are used. Here we see that the sharp negative slope that happens at approximately 30 singular values corresponds with the blurry images (SNR=28 dB, CR=4.25) for image size (256×256) that was shown in Fig.2b and (SNR=30 dB, CR=8.5). As we continue to reintroduce a greater number of singular values, we can see the quality of the image increase (SNR=37.4 dB, CR=1.4) as shown in Fig.2c, but we can see almost as many details with 90 singular values as we could see with the original 256sv Graph in Fig.3 show the variation of compression with different singular values (rank), and the relationship between the compression ratio (CR) and the number of singular values (rank) are inversely proportional to each other.

a. Original image(R=256)

b. Compressed image (R=30)

c. Compressed image (R=90)

Figure.2: Grayscale image (256×256) compression with different ranks

Table.1: Grayscale image (256×256) compression with different ranks

Rank	CR	Size (pixel)	Error	MSE	PSNR (dB)
30	**4.2583**	15390	6.3610×10^{-6}	97.0618	**28.2603**
60	2.1292	30780	1.9184×10^{6}	29.2721	33.4663
90	**1.4194**	46170	0.7710×10^{6}	11.7652	**37.4248**
120	1.0646	61560	0.3291×10^{6}	5.0216	41.1224
150	0.8517	76950	0.1301×10^{6}	1.9855	45.1521
180	0.7097	92340	0.0425×10^{6}	0.6486	50.0113
210	0.6083	107730	0.0083×10^{6}	0.1267	57.1020
240	0.5323	123120	0.0003×10^{6}	0.0041	71.9730

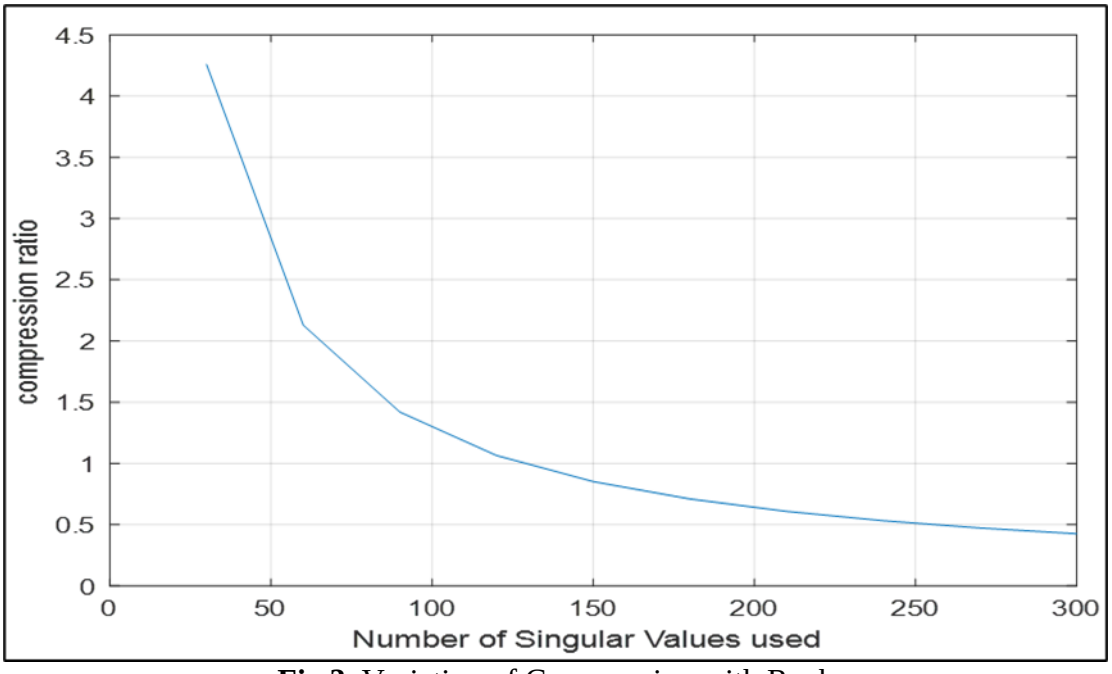


Fig.3. Variation of Compression with Rank

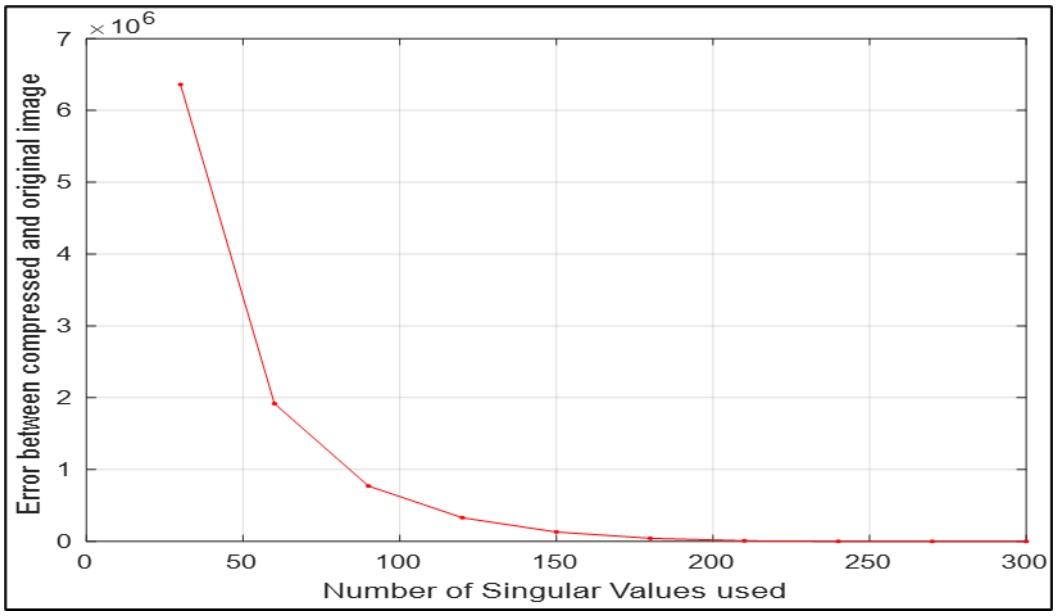


Fig.4. Variation of Error with Rank

Grayscale image compression using (SVD) with the same rank

We have taken four different images of the same sizes (512). The images are gray scale images, SVD is performed on them. The performance of SVD in these images vary based on the type of image. Four images that have been considered here are Birds (512x512), Lena (512x512), Baboon (512x512), and Scene (512x512). The accuracy of the image compression and reconstruction quality is measured by the PSNR and the Mean Square Error.

PSNR has been computed considering the maximum value of a pixel to be 255 in gray scale. A higher value for PSNR indicates a better reconstruction of the image. Tables 2, 3, 4, and 5 tabulate the image compression achieved through SVD for different rank values and its corresponding graph is shown in the Fig 5.

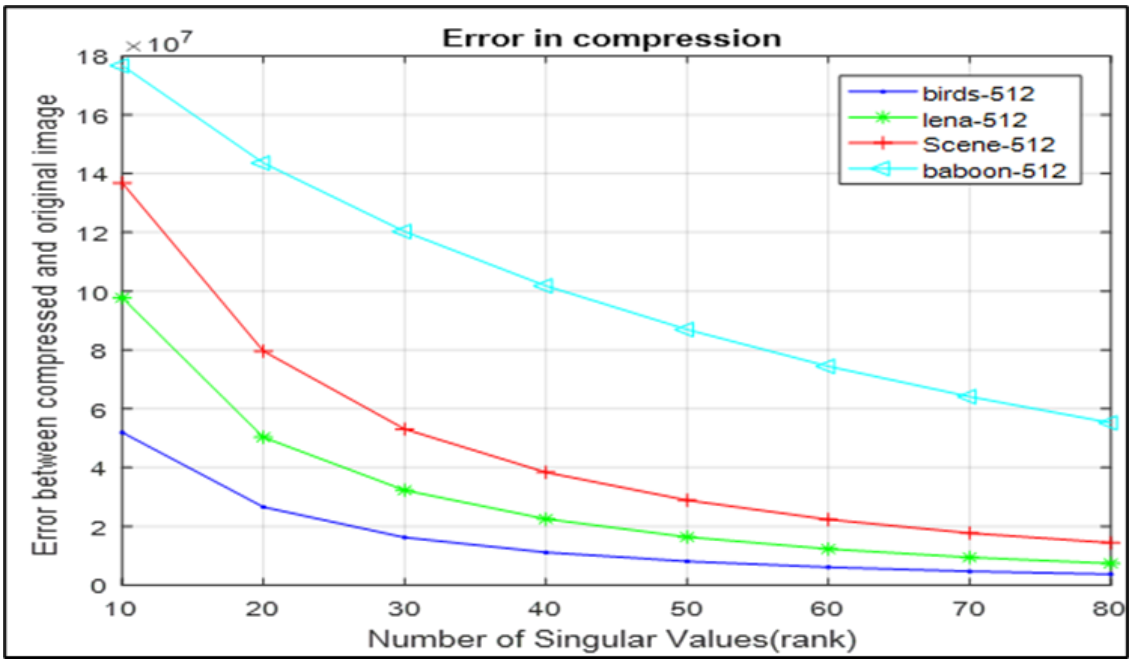


Fig. 5. Error of different images at the same rank**.**

Table 2. Results of compressed image (Lena) with SVD.

Test image Lena	Rank	Compression ratio	Error	MSE	PSNR
	10	25.5750	9.773 e+07	372.8268	22.4157
	30	8.5250	3.226 e+07	123.0864	27.2287
	50	5.1150	1.637 e +07	62.4584	30.1749
	70	3.6536	0.944 e+07	36.0199	32.5654
	90	2.8417	0.584 e+07	22.2886	34.6500

Table 3. Results of compressed image (baboon) with SVD.

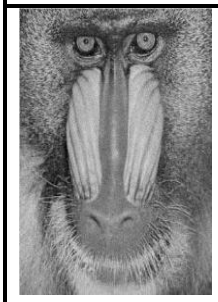
baboon	Rank	Compression ratio	Error	MSE	PSNR
	10	25.5750	1.7672e +08	674.1172	19.8434
	30	8.5250	1.2028e +08	458.8475	21.5141
	50	5.1150	0.8693e +08	331.6044	22.9246
	70	3.6536	0.6409e +08	244.4683	24.2486
	90	2.8417	0.4766e +08	181.8213	25.5344

Table 4. Results of compressed image (birds) with SVD.

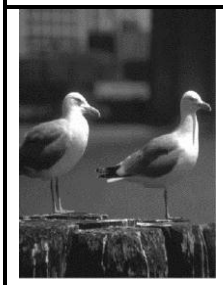
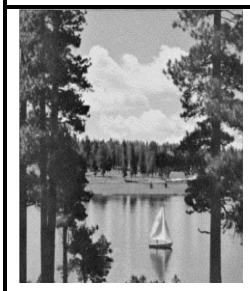
Birds	Rank	Compression ratio	Error	MSE	PSNR
	10	25.5750	5.1908e+07	198.0114	25.1639
	30	8.5250	1.6216e+07	61.8587	30.2168
	50	5.1150	0.8078e+07	30.8160	33.2430
	70	3.6536	0.4688e+07	17.8835	35.6063
	90	2.8417	0.2971e+07	11.3320	37.5877

Table 5. Results of compressed image (Scene) with SVD

	Rank	Compression ratio	Error	MSE	PSNR
	10	25.5750	1.3682e+08	521.9233	20.9547
	30	8.5250	0.5308e+08	202.4765	25.0671
	50	5.1150	0.2888e+08	110.1835	27.7096
	70	3.6536	0.1770e+08	67.5061	29.8374
	90	2.8417	0.1183e+08	45.1218	31.5869

Implementation to full color images using (SVD)

We have demonstrated compression for a grayscale image, now we will expand this process to full color images. For this simulation we chose a full color detailed image, Colored images have the same application for image compression, but it requires a few more calculations. The difference between a grayscale image and a colored image is that we need to store 3 bytes of information per pixel rather than 1 byte per pixel. This is because the red, green, and blue pixel values are now different rather than the same, so we have to represent each individually. First, we need to take a colored image, and split it into three new images, a red-scale, green-scale, and blue-scale image. We can treat the red-scale, green-scale, and blue-scale images just like we did with the grayscale image. This time the values 0 through 255 on our table represent only the saturation of that particular color. We can compute the SVD computation on each of these images separately, and then combine them back together to create our colored image. As we compute the SVD and only reintroduce specific singular values, we see the image quality increase. With only 10 singular values (rank=10), we have a very good idea of what we're looking at and we can recognize the compressed image from the original image. As singular values are increased to (rank= 90), we are able to see the image more clearly and very close to original. To compute the image sizes again, we have to add the matrix three times instead of just once as in a grayscale image. The error is a little more difficult to plot, as the graph would be three dimensional, since the image has three layers. The error for full color images is more complex to observe than the error corresponding to a grayscale image, due to the fact that we separated the color image into three separate color saturation matrices and the error of these three components is presented in Tables 6 and 7. The error curves in Fig.6 and 7 represent the accumulated error when comparing each modified color saturation matrix to the corresponding color saturation matrix from the original image. We can again observe a significant change in sharpness from the compressed images using relatively few singular values. In addition to reducing the apparent error by adding more singular values, we also notice a significant difference between the error curves within each pixel color.

Table 6. Error of the color image components (Red, Green and Blue)

	Rank	CR	Error		
			Red comp	Green comp	Blue comp
 peppers 512x1536	10	38.3813	8.4737e+07	1.8133e+08	9.6511e+07
	30	12.7938	2.7402e+07	0.4785e+08	2.6735e+07
	50	7.6763	1.4954e+07	0.2296e+08	1.3955e+07
	70	5.4830	0.9909e+07	0.1424e+08	0.9240e+07
	90	4.2646	0.7214e+07	0.0979e+08	0.6750e+07

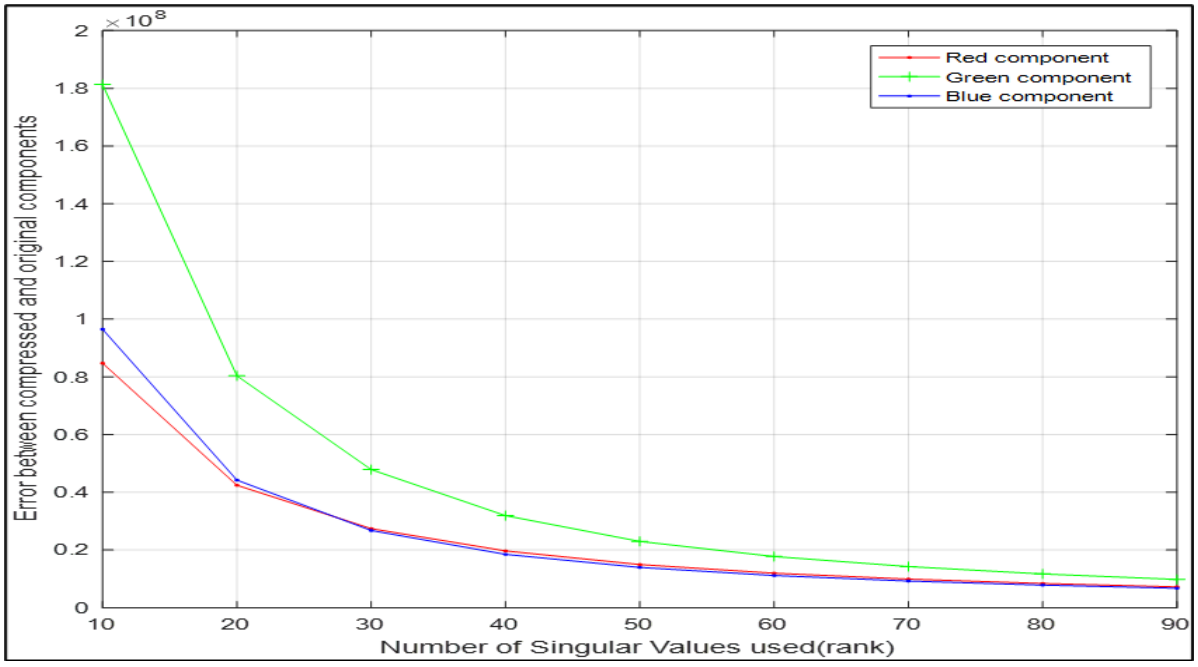


figure 6 – Error curves for the red, green, and blue pixel saturation levels from the peppers image.

Table 7. Error of the color image components (Red, Green and Blue)

Fruits	Rank	CR	Error		
			Red comp	Green comp	Blue comp
 480 x1536	10	36.5533	3.8613e+07	5.1176e+07	1.0640e+08
	30	12.1844	1.3167e+07	2.1119 e+07	0.5281e+08
	50	7.3107	0.7322e+07	1.1789e+07	0.3063e+08
	70	5.2219	0.4374e+07	0.6844e+07	0.1845e+08
	90	4.0615	0.2699e+07	0.4073e+07	0.1131e+08

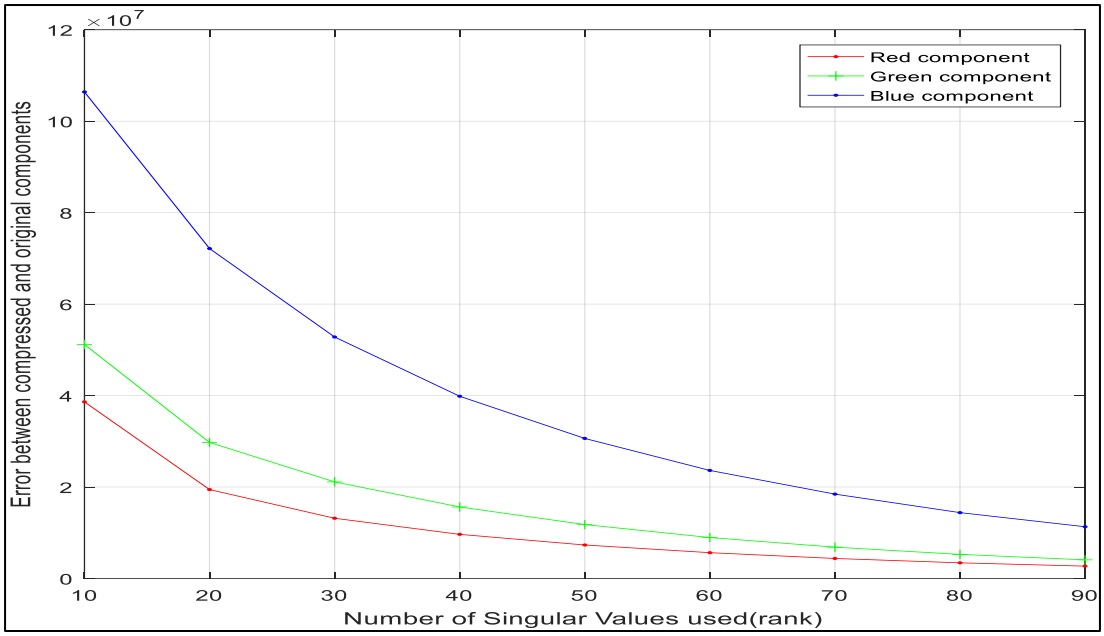


Figure 7– Error curves for the red, green, and blue pixel saturation levels from the fruits image.

Conclusion

By applying the process of Singular Value Decomposition to images, we have isolated the least important pieces of information that are stored in the images and have removed them methodically, leaving only the most important components of the images. This process of removing the smallest singular values from the saturation matrices allows us to retain as much of the image quality as possible. The approach applied to a gray scale image and also the colored images (RGB matrices). The performance of SVD varies depending on the type of image being compressed and also depending on the number of singular values (rank). The quality of the reconstructed image increases with high rank values but it results in lesser compression. Singular Value Decomposition (SVD) is a simple, robust and reliable technique. This SVD technique provides stable and effective method to split the image matrix into a set of linearly independent matrices. SVD provides good compression ratio and also a practical solution to image compression problem. The results clearly display the compressed outputs for different rank (r) values. Thus, selection of r value plays a crucial role in this SVD based image compression technique.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Acharya, T., & Tsai, P. S. (2005). JPEG2000 standard for image compression. Wiley-Interscience.

[2] Almara, H., & Barhoum, K. (2011). Grayscale image compression based on min max block truncating coding. IJCSI International Journal of Computer Science Issues, 8(6), 1–6.

[3] Andrews, H. C., & Patterson, C. L. (1976). Singular value decomposition (SVD) image coding. IEEE Transactions on Communications, 24(4), 425–432.

[4] Anusha, B., Soujanya, Y., & Vyshnavi, E. (2018). Image compression using SVD on LabVIEW with vision module. International Journal of Computational Intelligence Research, 14(1), 59–68.

[5] Archana, D. (2013). Analysis of image compression methods based on transform and fractal coding [MTech thesis].

[6] Chadi, A., & Doaa. (2011). Image compression using block truncation coding. Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), February Edition.

[7] Cho, N. I., & Lee, S. U. (1990). DCT algorithms for VLSI parallel implementation. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(1), 121–127.

[8] Delp, E. J., & Mitchell, O. R. (1979). Image compression using block truncation coding. IEEE Transactions on Communications, 27(9), 1335–1342.

[9] Dhumal, P., & Deshmukh, S. (2016). Survey on comparative analysis of various image compression algorithms with singular value decomposition. International Journal of Computer Applications, 133(6), 33–38.

[10] El Abbadi, N. K., & Al Rammahi, A. (2014). Image compression based on SVD and MPQ-BTC. Journal of Computer Science, 10(10), 2095–2104.

[11] ElAsnaoui, K. (2020). Image compression based on block SVD power method. Journal of Intelligent Systems, 29(1), 1345–1359.

[12] Franti, P., Nevalainen, O., & Kaukoranta, T. (1994). Compression of digital images by block truncation coding: A survey. The Computer Journal, 37(4), 308–332.

[13] Gonzalez, R. C., & Woods, R. E. (1993). Digital image processing. Addison-Wesley.

[14] Huang, Y. (2022). Overview of research progress of digital image processing technology. Journal of Physics: Conference Series, Tongji University.

[15] Hussien, A. (2016). Color image compression by using absolute moment block truncation coding and hadamard transform. International Journal of Computers and Technology, 15(7).

[16] Li, H., Wang, S., & Wen, Q. (2005). A novel watermarking algorithm based on SVD and Zernike moments. International Conference on Intelligence and Security Informatics, 448–453.

[17] Mounika, K., & Navya, D. S. (2015). SVD based image compression. International Journal of Engineering Research and General Science, 3(2).

[18] Naidu, S. V., & Yahia, H. (2015). A comparison of wavelet based techniques for resolution enhancement of moderate resolution satellite images. National Conference on Recent Advances in Electronics & Computer Engineering (RAECE), 263–266.

[19] Narayan, T., & Ashok, L. (2016). Digital image processing techniques - Survey. International Multidisciplinary Research Journal, 5(11).

[20] Priya, K., & Banu, R. B. (2019). Digital image processing techniques - a review. Journal of Emerging Technologies and Innovative Research (JETIR), 6(9).

[21] Sandhu, K., & Singh, M. (2018). Image compression using singular values decomposition (SVD). International Journal of Latest Research in Science and Technology, 7(5), 5–8.

[22] Saua, K., & Kumar, R. (2013). Image compression based on block truncation coding using Clifford algebra. International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA).

[23] Shu, X., & Wu, X. (2013). Image enhancement revisited: From first order to second order. IEEE International Conference on Image Processing, 886–890.

[24] Somasundaram, K., & Vimala, M. S. S. (2010). Efficient block truncation coding. International Journal on Computer Science and Engineering, 2(6), 2163–2166.

[25] Swathi, H., & Sohini, S. (2017). Image compression using singular value decomposition. IOP Conference Series: Materials Science and Engineering.

[26] Vasanthakumari, A. (2015). Image compression techniques. IJARIIE, 1(5).

[27] Venkataseshaiah, B., & Roopadevi, K. N. (2016). Image compression using singular value decomposition. International Journal of Advanced Research in Computer and Communication Engineering, 5(12).

[28] Vimala, S., & Uma, P. (2011). Improved adaptive block truncation coding for image compression. International Journal of Computer Applications, 19(7).

[29] Wallace, G. K. (1992). The JPEG still image data compression standard. IEEE Transactions on Consumer Electronics, 38(1), xviii–xxxiv.

[30] Wang, Z., & Bovik, A. C. (2004). Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600–612.

[31] Zheng, W., & Liu, Y. (2011). Research in a fast DCT algorithm based on JPEG. International Conference on Consumer Electronics Communications and Networks.

