Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

IMAGE COMPRESSION USING SINGULAR VALUE DECOMPOSITION

Priyanka Malgaonkar, Assistant professor, department of humanities and applied sciences, Atharva College of Engineering, Malad.
Deepika Panchal, Assistant professor, department of humanities and applied sciences, Atharva College of Engineering, Malad.
Divya Acharya, Assistant professor, department of humanities and applied sciences, Atharva College of Engineering, Malad.
Vinayak Sawant, Assistant professor, department of humanities and applied sciences, Atharva College of Engineering, Malad.

Abstract: Image compression is the process of reducing the size of an image file while retaining its quality. When we compress a digital photograph or graphic file, it maintains the same image resolution but shrinks the amount of processing data that a computer uses to view or display that image which ultimately reduces the memory used for the storage of image files. There are various algorithms used for image compression. Linear Algebra (SVD) plays an important role in image compression. In this paper, we will discuss what is Singular Value Decomposition (SVD), how to compute singular value decomposition(SVD) and the size of stored images is reduced by removing small singular values. We will use MATLAB to get the final output.

Keywords: *Image compression, Singular Value Decomposition, MATLAB*

I Introduction

To store a large amount of data much storage space is required. We often share information through pictorial data but many times we are running out of storage space and the speed of our devices get reduced. Image compression aims to minimise the amount of data needed to represent an image. The algorithm for image compression's primary objective is to represent images using the fewest number of bits.

In order to save an image using less memory while maintaining the image quality, we explore employing the technique of Singular Value Decomposition to compress the size of the saturation matrices while keeping the most crucial elements..[3][6]

II Singular Value Decomposition: What is it?

To perform singular value decomposition (SVD), a matrix A must be transformed into the form $A = U\Sigma V^T$ With the use of this computation, we are able to keep the crucial singular values that the image needs while letting go of the values that are not as crucial to maintaining the image's quality.
The square roots of the eigenvalues of the n × n matrix A^TA, which are normally arranged by magnitude in decreasing order, make up the singular values of a m × n matrix A.

How to compute the SVD of Matrix

we solve one example of SVD using the theory process. Let $A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$

We begin by forming matrix AA^T

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

Next step is to calculate Eigenvalues of AA^T. For that compute determinant of $(AA^T - \lambda I)$. Find characteristic equations i.e. det $(AA^T - \lambda I) = 0$.
After solving we will get the characteristic equation as $\lambda^2 - 34\lambda + 225 = 0$.
The roots of this equation are $\lambda = 25, 9$.
Therefore the singular values are $\sigma_1 = 5$, $\sigma_2 = 3$. Now we have to form matrix V.
We find orthonormal set of eigenvectors of A^TA. The eigen values of A^TA are 25, 9 and 0.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

For $\lambda = 25$ we have, $A^TA - 25I = \begin{bmatrix} -12 & 12 & 2 \\ 12 & -12 & -2 \\ 2 & -2 & -17 \end{bmatrix}$

We then solve the homogeneous equation $(A^TA - 25I)\, X_1 = 0$. We get $X_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

We then normalize the eigenvector by dividing by its magnitude. We get $v_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix}$

We do this for each eigenvalue and form a matrix $V = [v_1 \quad v_2 \quad v_3] = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{18} & 2/3 \\ 1/\sqrt{2} & -1/\sqrt{18} & -2/3 \\ 0 & 4/\sqrt{18} & -1/3 \end{bmatrix}$

We list non-zero singular values down the major diagonal in decreasing order to decide matrix Σ.
Hence $\Sigma = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$.

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

We can compute U by the formula $u_i = \frac{1}{\sigma} A v_i$. This gives $U = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$.

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

III Application to Image compression

When we look at the white colour on our screen, there is no white pigment on the screen. It is a combination of red, green, and blue colours that are depicted on the screen by minuscule pixels. When viewed from a distance, the saturation of each pixel in this grid-like pattern causes it to appear a distinct colour. These red, green, and blue pixels range in saturation on a scale of 0 to 255; with 0 being completely off, and 255 being completely on.[2] A picture can represent data in a matrix because of the pixel's grid-like nature.

Consider a grayscale image. Red, green, and blue values must match in order to make an image grey. We can represent a pixel as having a value of 0 through 255 and then repeat that value across the red, green, and blue saturation to get the corresponding shade of grey.[2]

Let's say we have a grayscale image with a size of 5184 x 3456 pixels. A matrix that is also 5184 x 3456 and with values ranging from 0 to 255 can be used to represent each of those pixels; the matrix's dimensions are5184 x 3456. Now, we should have to keep track of exactly 5184 x 3456 digits if we want to store that image. If it was coloured, it would be triple that of a grayscale image, which equals 1.64 MB for a grayscale image or 4.92 MB for a coloured image. To save memory on the image we can compute singular value decomposition and then calculate some level of precision.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

a) Implementation in Grayscale image

In MATLAB, we use and modify existing code from Dr Brady Matthews' paper "Image Compression using Singular Value Decomposition" to load an image, isolate the corresponding saturation matrix, and then modify the matrix based on its singular values[2][3].

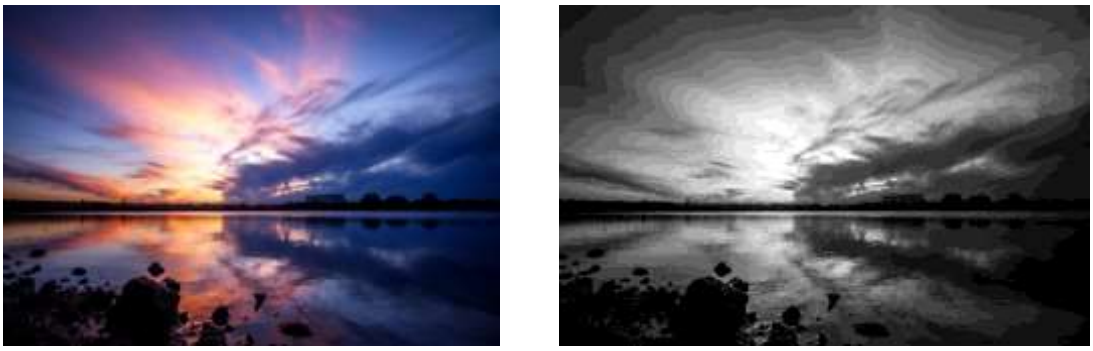


Fig. 1 Coloured and Grayscale image.[4]

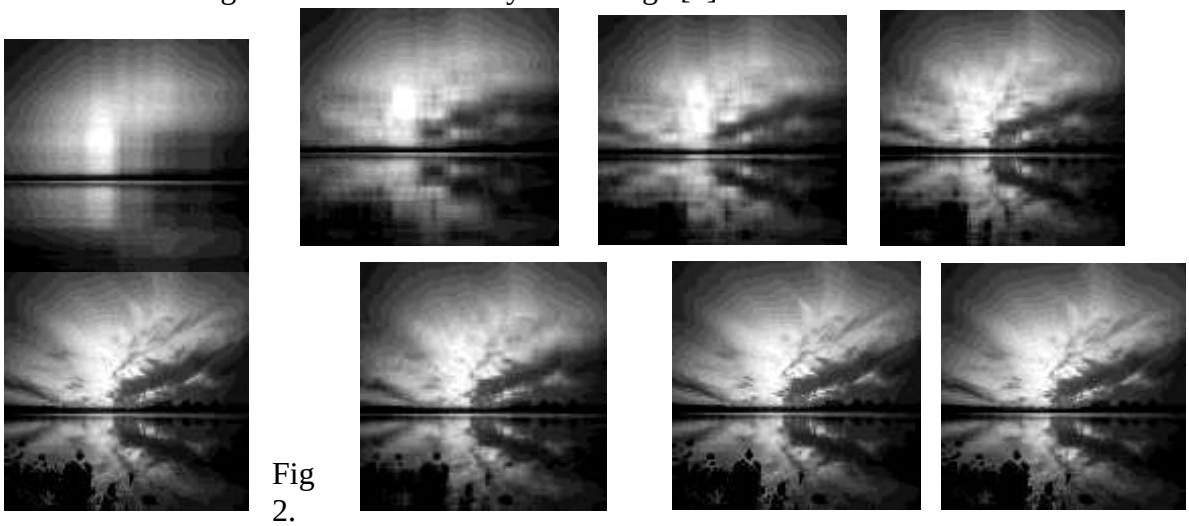


Fig 2.

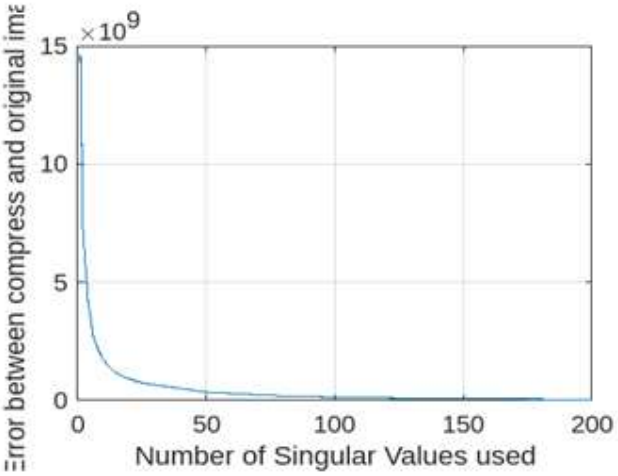
Images formed using MATLAB with singular values {{2,6,12,20}, {25,50,100,150} }

From fig 1. And 2 we get the following results:

Original Image	689 KB (7,06,400 bytes)
with a singular value of 50	440 KB (4,51,239 bytes)
with a singular value of 100	480 KB (4,92,335 bytes)
with a singular value of 150	500 KB (5,12,000 bytes)

This shows that these modes save quite a bit of memory. The inaccuracy in the picture compression process and how the image differs from the original image are displayed in the following graph. The error is calculated as the difference between our new image and our original image, and this difference is then plotted on a graph. We can infer from figs. 3 and 4 that when more singular values are used, the rate change of the error loss becomes less significant. [3]

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023



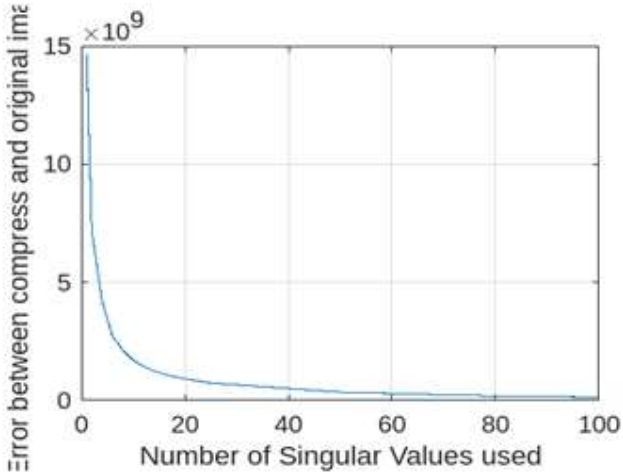


Fig.3 By taking singular values up to 200 Fig. 4 By taking singular values up to 100

b) *Implementation in colour image*

As Shown above we used image compression for the grayscale image. Now we will expand this process for a colour image. For this, we choose a colour image of the flower as shown in fig. 5

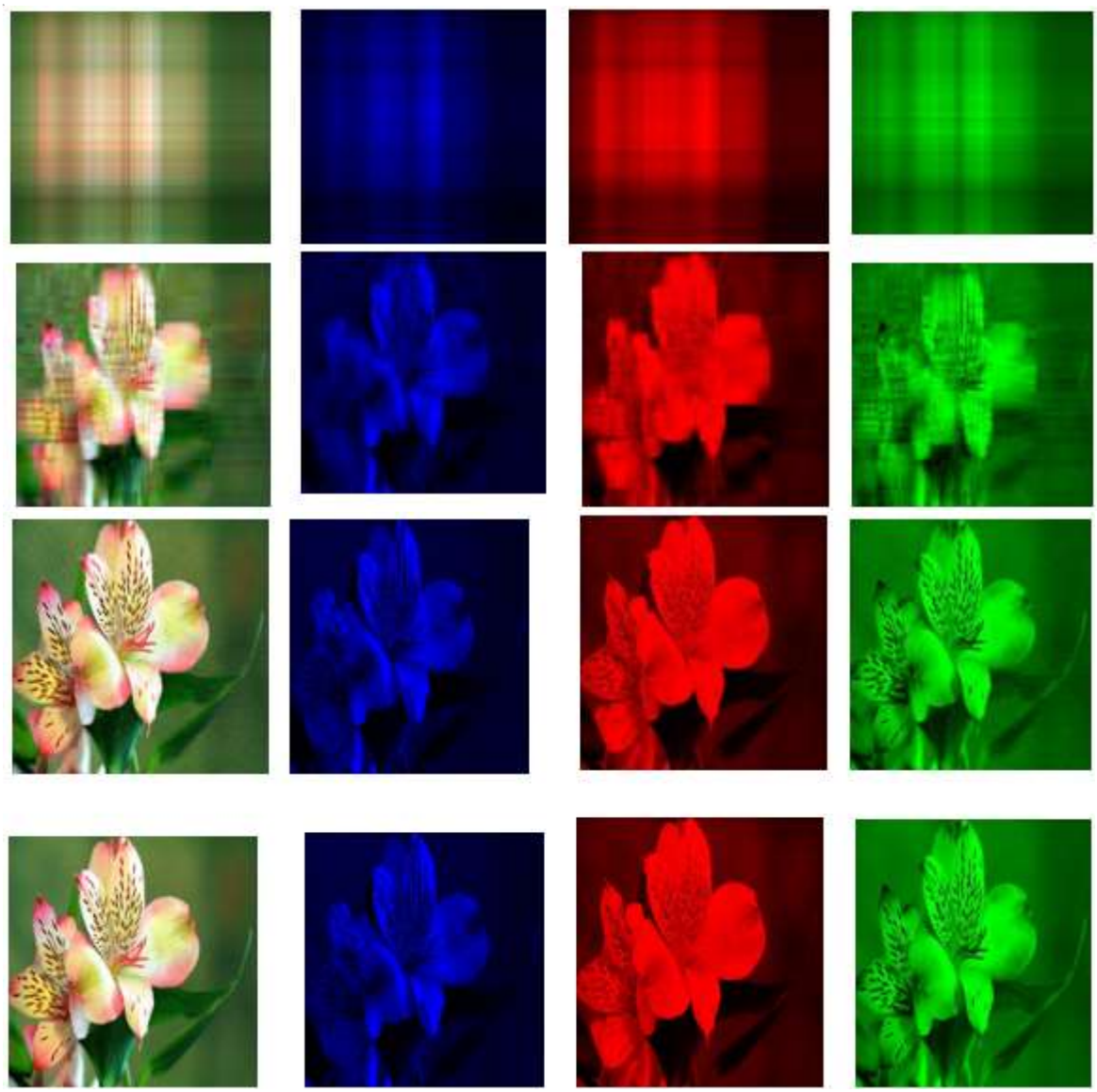
Fig. 5 [5]

Each pixel in the full colour image has colour saturation representation values of 0 to 255 for red, green, and blue. Since the image becomes more complicated as a result, more storage space is needed to save it. We can determine the contribution each colour makes to each pixel by comparing how each one is represented to the full-colour image, as seen in fig 6

Fig. 6

Since each of these three colours contains a unique matrix of image information, we must first divide the full-colour image into its red, green, and blue layers before we can perform the SVD method. Each of the colour matrices will have its smallest singular values removed, and we'll use those values to modify the matrices before reconstructing the full-colour image. The quality of the image in Fig. 7 improves as we compute the SVD and only reinsert particular singular values. There isn't much we can tell about the original image from a value with just one single. We can more easily identify the image as a flower as singular values are added back in [2].

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023



Compressed Image

If we compared the original and compressed images, we can see a noticeable difference in storage size. i.e our main challenge of saving storage space is fulfilled. The original image took 1.64 MB of memory and the compressed image with 100 singular values is taking 775 KB of memory which is almost 50% of the original image. Comparing them side by side, they appear to be nearly identical, but their storage capacities and information matrices are substantially less. Instead of naturally lowering the number of pixels in these comparisons, we have kept the original amount by removing unimportant singular values.

Conclusion:

We apply singular value decomposition to both greyscale and colour image and from above result we can say that, our main aim to compressed data without reducing the quality of an image is achieved. For the colour image, when we apply singular value 100, we are able to save half of the memory. In the future, we can apply this process to every frame of videos will definitely save significant amount of storage. This process will also help in face recognization.

References:

1) Lay, David C., Linear Algebra and its Applications, 3rd updated Edition, 2005
2) Mathews, Brady, "Image Compression using Singular Value Decomposition (SVD)," 2014.
3) Elizabeth A. Compton and Stacey L. Ernstberger, PhD, " Singular Value Decomposition: Applications to Image Processing", 2020
4) Gaston Abascal, "Tigre, Buenos Aires, Argentina" https://unsplash.com/s/photos/Tigre%2C-Buenos-Aires%2C-Argentina, 2018.
5) Anthony Pexels, "pexels-anthony-)-132468.jpg", 2020.
6) H R Swathi, Shah Sohini, Surbhi and G Gopichand, "Image compression using singular value decomposition"