PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER

UE18MA251- LINEAR ALGEBRA

MINI PROJECT REPORT

ON

IMAGE COMPRESSION USING SINGULAR VALUE DECOMPOSITION (SVD)

Submitted by:

1. **Name:** SAMPREETH NAIK **SRN:** PES2201800179
2. **Name:** NITISH S **SRN:** PES2201800368
3. **Name:** SUPREET RONAD **SRN:** PES2201800705

Branch & Section: CSE, 'A'

PROJECT EVALUATION

(For Official Use Only)

Sl.No.	Parameter	Max Marks	Marks Awarded
1	Background & Framing of the problem	4	
2	Approach and Solution	4	
3	References	4	
4	Clarity of the concepts & Creativity	4	
5	Choice of examples and understanding of the topic	4	
6	Presentation of the work	5	
	Total	25	

Name of the Course Instructor : Dr.Girish.V.R.

Signature of the Course Instructor :

TABLE OF CONTENTS

1. INTRODUCTION

It is well known that the images, often used in a variety of computer applications, are difficult to store and transmit because these images require a large memory space and bandwidth. There are many techniques to overcome this problems:

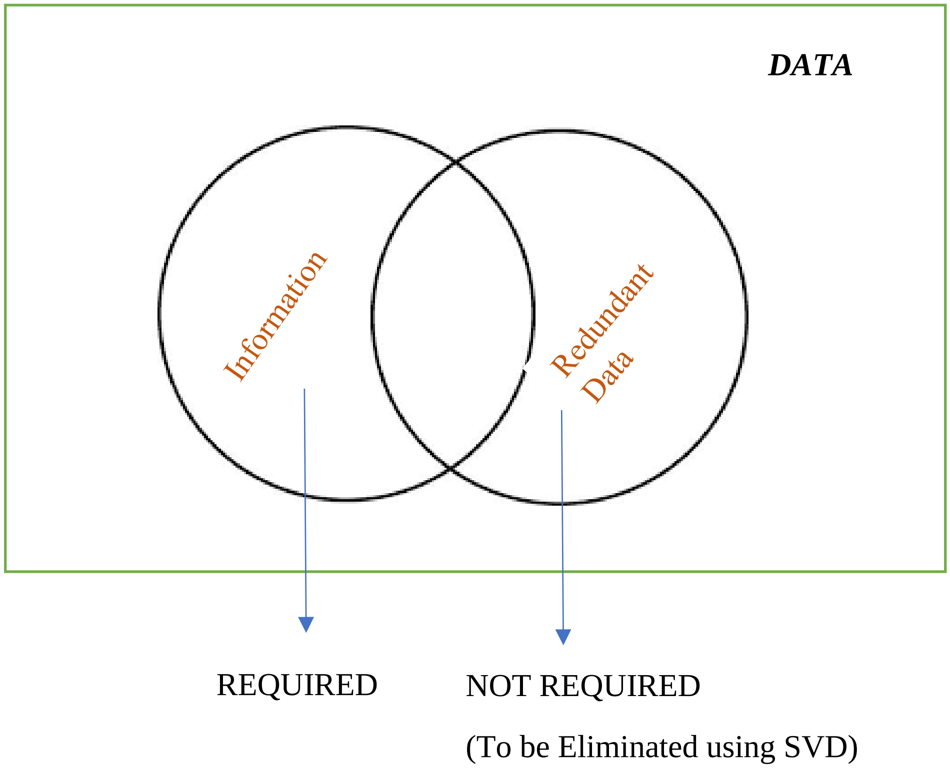
1. The most widely used image compression technique today is JPEG (Joint Photographic Experts Group) which uses ***DCT (Discrete Cosine Transform)*** for compression of images.
2. Another possible solution discussed in this paper is to use a data compression technique where an image can be thought of as a matrix and then the operations are performed on the matrix. Here, Image compression is achieved by using ***Singular Value Decomposition (SVD)*** technique on the image viewed as a matrix.

Advantages of using SVD :

- property of energy compaction
- ability to adapt to the local statistical variations of an image
- can be performed on any arbitrary, square, reversible and non reversible matrix of m x n size.
- The psycho visual redundancies in an image are used for compression. Thus an image can be compressed without affecting the image quality.
- The MSE and compression ratio are used as thresholding parameters for reconstruction. SVD is applied on a variety of images for experimentation.

into 3 matrices:

The basic idea of SVD is taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space so that it exposes the substructure of the data more clearly and orders it from the most variation to the least.



Singular Value Decomposition (SVD) is said to be a significant topic in linear algebra by many renowned mathematicians. SVD has many practical and theoretical values, other than image compression. One special feature of SVD is that it can be performed on any real (m,n) matrix. It factors any given matrix say, A into three matrices U, S, V, such that, $A = USV^T$. Where U and V are orthogonal matrices and S is a diagonal Matrix.

If A is a matrix of order m*n and rank ‘r’ , then it can be factorized

$$\boxed{A_{m*n} = U_{m*m} . S_{m*n} . V^T_{n*n}}$$

Properties of SVD :

1. The rank of matrix A is equal to the number of non-zero/singular values.
2. $AA^T = U.S^2.U^T$ and hence U diagonalizes the matrix A and column vectors of U are the eigenvectors of AA^T.
3. Similarly, $A^TA = V.D^2.V^T$, V diagonalizes A and hence eigenvectors of A^TA are column vectors of V.
4. The singular values of matrix A i.e. σ1, σ2, σ3……….., σN are unique, however the matrices U and V are not unique.

Brief Method of implementing Image Compression using SVD:

1) The Singular Value Decomposition expresses image data in terms of number of eigen vectors depending upon the dimension of an image.
2) The work is concentrated to reduce the number of eigen values required to reconstruct an image.
3) SVD is a linear matrix transformation used for compressing images. Using SVD an image matrix is represented as the product of three matrices U, S, and V where S is a diagonal matrix whose diagonal entries are singular values of matrix A.
4) The image A can also be represented by using less number of singular values, thus, presenting necessary features of an image while compressing it.
5) The compressed image requires less storage space as compared to the original image.
6) To choose the value of k i.e. number of Eigen values for compression and reconstruction of the image is an important decision for acceptable reconstruction. It varies with application and in this work compression ratio is used to select the number of Eigen values out of the maximum. It is observed that if the value of k chosen is equal to the rank of the image, the reconstructed image is closer to the original image. As the value of k decreases from the rank image quality degrades. Second observation is that as the compression ratio is high, image quality is poorer and if compression ratio is low, image with superior quality can be reconstructed but with less compression. Therefore compression ratio and image quality is required to select appropriately.

2. REVIEW OF LITERATURE

- S K Singh et al. [1] has implemented compression of image. Image matrix is processed using the technique of Singular Value Decomposition (SVD). This technique carries out the compaction according to the compaction of energy. Moreover, it concentrates on the initial few columns which tend to have the localized content of the matrices. Overall energy of the image is represented by singular values of the image. Storage and transmission of image demands compression to be carried out on the original image. To resolve this problem, image is seen as a matrix and compression of the matrix is performed by performing various operations on the matrix. The SVD can be implemented on matrices of varying sizes and types such as arbitrary, square, reversible and non-reversible matrices. SVD reduces the storage space of an image. Effect of rank is also shown in SVD decomposition that will judge the quality of compressed images.

- M. E. I. Tian et al. [2] has proposed three techniques for compressing the images. They have also studied and analyzed the efficiency of SVD based image compression techniques. This paper has introduced a technique wherein efficient representation of singular vectors in each sub-block is used. This approach is called an adaptive singular value selection scheme. Three schemes are-Direct compression and decomposition scheme, adaptive singular value selection scheme, singular value subtracting one update scheme. In the Direct scheme, the original matrix is approximated using singular values which are fewer. As opposed to compressing the entire image in one go, this selection scheme tries to divide the original image into sub-block which are smaller in size with the goal of working with these smaller images. This approach is chosen with the aim of using the uneven complexity present in the original image to our advantage.

- P. Wadem et al. [3] has proposed changing the hybrid KLT-SVD system to KTL coding system. KLT stands for Karhunen-Lobve transform. An image contains variations in the statistics which are local to the image. We can exploit this fact and use the transformation adaptive technique to transform the coding of images. This technique uses the similarities between SVD and KLT and due to this coding efficiency of the technique is also improved. If the purpose is to use the transform on all the blocks which build the image, KLT can be used efficiently. However, improved compression for a given block is achieved by SVD. So in order to use this advantage, a system is built that forms the SVD transforms and creates the approximations using them. Lagrange technique can be used to reduce the distortion of the image during reconstruction. This can be used for switching to KLT. But the limitation here is the number of bits used for reconstructing every block is very high. This increases the cost associated with this technique. This can be improvised by using substitute methods.

- T. J. Peters et al. [4] has implemented SVD (singular value decomposition) to compress the microarray image. Huge amounts of DNA information for research purposes are stored as microarray images. These are of high resolution images which highlights minute details of the image. Because of the high resolution, these images tend to be larger in size, which means storage on the hard disk requires a lot of space. So it is very important to reduce the size of the image without compromising the quality or compromising the amount of detail present in the image. This calls for a comparatively complicated process, where microarray images need to be clustered and classified before selecting the features. SVD can be used here to divide the image into small sub -images and on each sub-image SVD is performed. This method gives a better high peak signal to noise ratio in addition to increasing compression ratio.
- H. S. Prasantha et al. [5] have implemented another approach for compression of images using singular value decomposition (SVD). By applying singular value decomposition on the image matrix, compression of image is achieved .SVD takes into consideration. This method employs a data compression technique, wherein operations are performed on the matrix after the image is decomposed into matrices. For every image, matrix is found and as per points graph is plotted for dissimilar ranks in [6]. Further, it is observed that as

rank increases in image matrix, the number of entries will also increase which leads to make picture quality better correspondingly. Due to this, more compression ratio has been achieved by smaller ranks.

3. REPORT ON THE PRESENT INVESTIGATION

The most basic and simple application of SVD for image compression involves selecting the principal components to reproduce an approximated version of the image. Such kind of application involves a limit over the no. of components we may take. SVD transform decomposes the image A of size $m \times n$ into 3 matrices U (matrix of size $m \times m$ containing eigenvectors of A^TA), S (diagonal matrix of size $m \times n$ containing singular values in decreasing order) and V (matrix of size $n \times n$ containing eigenvectors of AA^T). Hence the overall size increases with that of originally being $(m \times n)$ to $(m \times m + n \times n + n\ (n < m))$. Hence the compression ratio may be defined as $(m \times n)$ / $(m \times k + n \times k + k)$ for the k principal components. For compression to be effective, this ratio must be greater than 1. We can derive an expression for the no. of maximum components k we may select so that the size of the selected components remains less than the original image as shown below. $(m \times n)\ /[\ k \times (m + n + 1)]\ >\ 1\ or\ k < (m \times n)/\ m + n + 1$ (1).

This expression is traditionally being used for the SVD based compression ratio and the maximum rank permissible for the purpose of image compression. But since the eigenvectors are orthogonal and unitary, we do not need to store all the values of the eigenvectors. If we make use of the orthogonal property of the eigenvectors, we can recover the values even if for the i^{th} vector, we skip i-1 values and store other values of the eigenvectors. For the first vector, we store all the values and hence we skip the upper triangular values of the eigenvectors matrices U and V. Since vectors are orthogonal, these values can be easily recovered by solving the system of linear equations thus formed as shown below:

$$\boldsymbol{A^1 \times A^2 = 0}$$

$$\boldsymbol{A^1 \times A^3 = 0\ and\ A^2 \times A^3 = 0}$$

$$\boldsymbol{A^1 \times A^4 = 0, A^2 \times A^4 = 0\ and\ A^3 \times A^4 = 0}$$

$$\boldsymbol{A^1 \times A^k = 0, A^2 \times A^k = 0, A^3 \times A^k = 0, \ldots A^{k-1} \times A^k = 0}$$

Above shown linear system of equations needs to be solve having one, two, three and $k-1$ unknowns respectively. In the similar way, we can calculate other values also for all the vectors. Note that there will be k linear equations for $k-1$ unknown value in case of the last kth vector. Since we can skip these values, the expression for the compression ratio and maximum rank needs to be redefined. Number of truncated values for each matrix of eigenvectors can be stated as $(k/2) \times (k-1)$. Total no. of truncated values is, therefore $k \times (k-1)$. New expression for compression ratio can be derived as shown:

$$\mathbf{CR'} = \frac{\mathbf{m * n}}{k(m+n+1) - k(k-1)} = \frac{\mathbf{m * n}}{k(m+n-k+2)}$$

Where, CR' is modified expression for compression ratio. Now as we have a new expression for the compression ratio, expression for maximum permissible rank needs to be redefined. It can be derived by solving the inequality shown below:

$$\frac{\mathbf{m * n}}{k(m+n-k+2)} > 1$$

$$k^2 - (m+n+2)k + (m \times n) > 0$$

Solving the inequality, we can derive the expression for maximum permissible rank for the purpose of image compression as shown below:

$$k < (1/2)\ \{(m+n+2) - \sqrt{[(m+n+2)^2 - 4 \times m \times n\ ]}\ \}$$

Now since we are skipping some values there must be some gain in the newly formed expression for compression ratio as compared to the previous expression. An expression for the gain can also be derived by subtracting the previous expression from the newly formed expression.

$$Gain = CR' - CR$$

$$Gain = \frac{\mathbf{m * n}}{k(m+n-k+2)} - \frac{\mathbf{m * n}}{k(m+n+1)}$$

$$= \frac{\mathbf{m * n}}{\mathbf{k}} \quad \{m + n + 1 - (m + n - k + 2) / (m + n - k + 2)(m + n + 1)\}$$

$$= \frac{\mathbf{(m * n)\ (k - 1)}}{k(m + n - k + 2)(m + n + 1)}$$

Clearly, storing only the lower triangular values along with principal diagonal is sufficient to recover the all values for the eigenvectors. And it contributes in the form of an increase in the compression ratio of the SVD based compression.

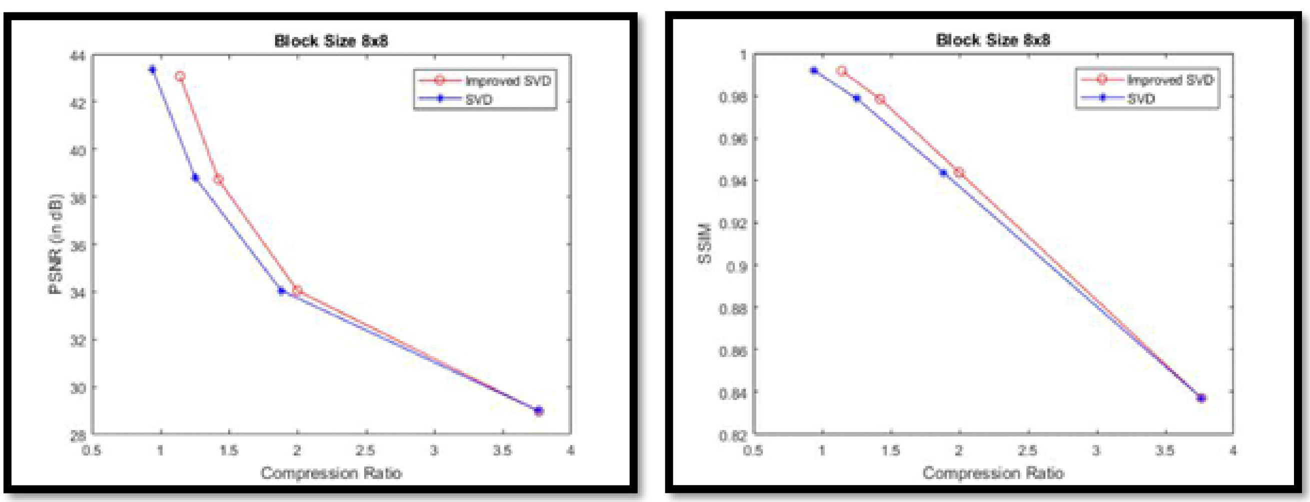


Fig.1 compression ratio vs PSNR and compression ratio vs SSIM graph plot for 8x8 block SVD.

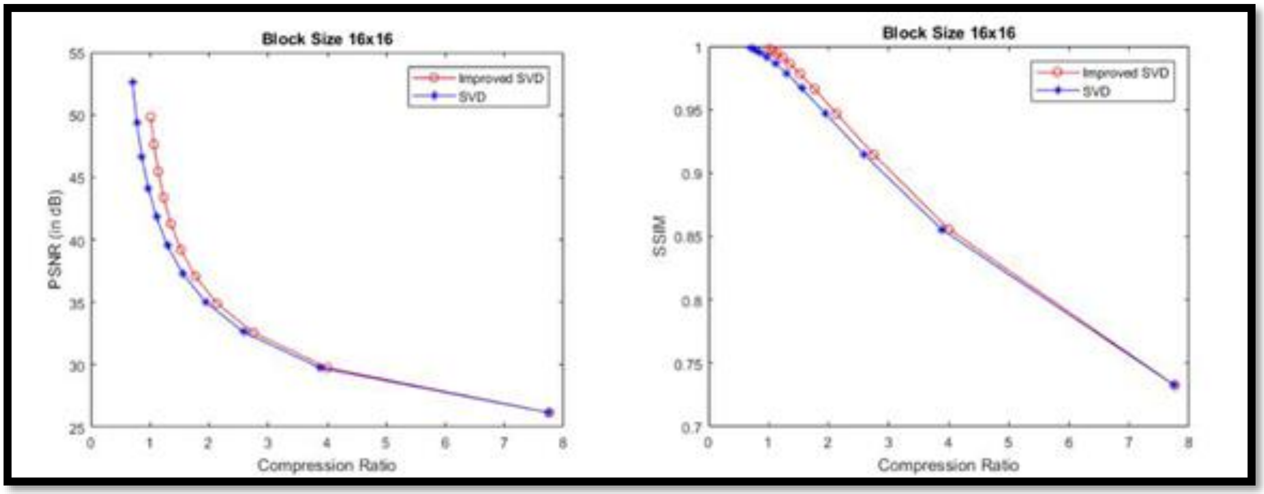


Fig.2 compression ratio vs PSNR and compression ratio vs SSIM graph plot for 16x16 block SVD.

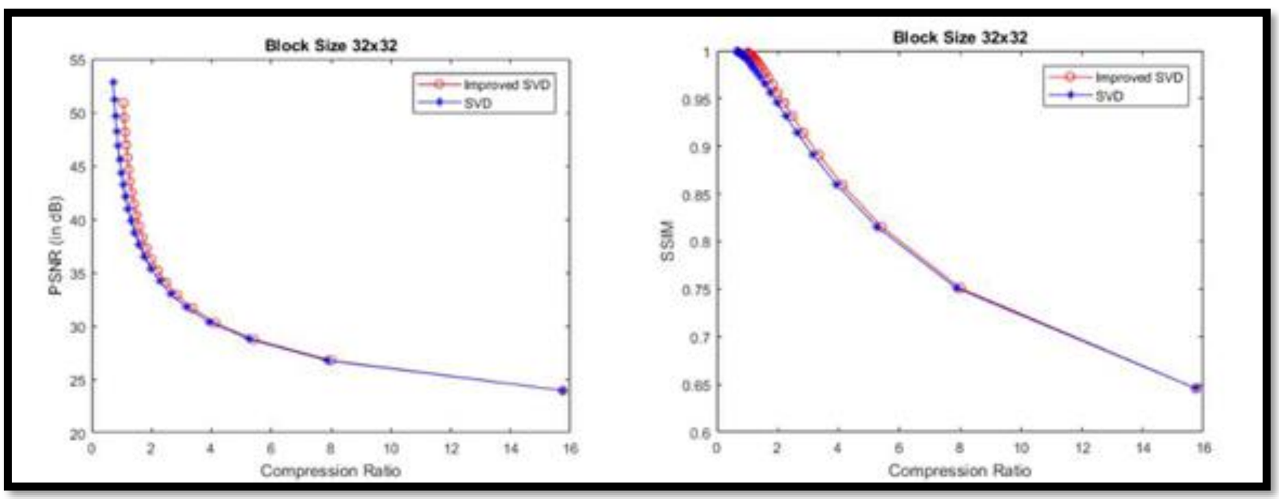


Fig.3 compression ratio vs PSNR and compression ratio vs SSIM graph plot for 32x32 block SVD.

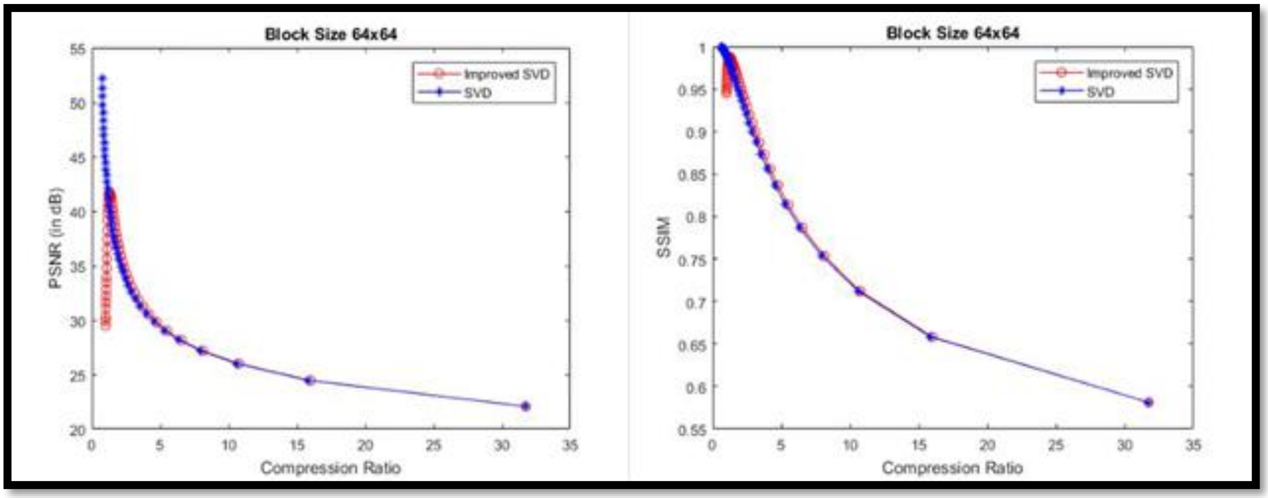
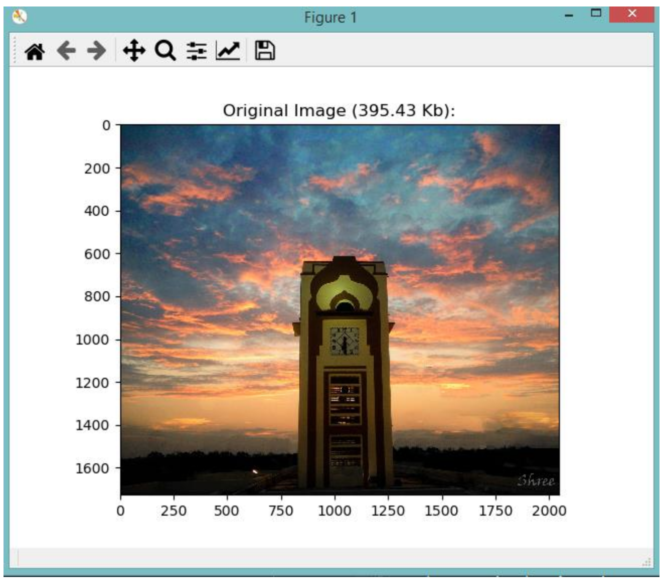


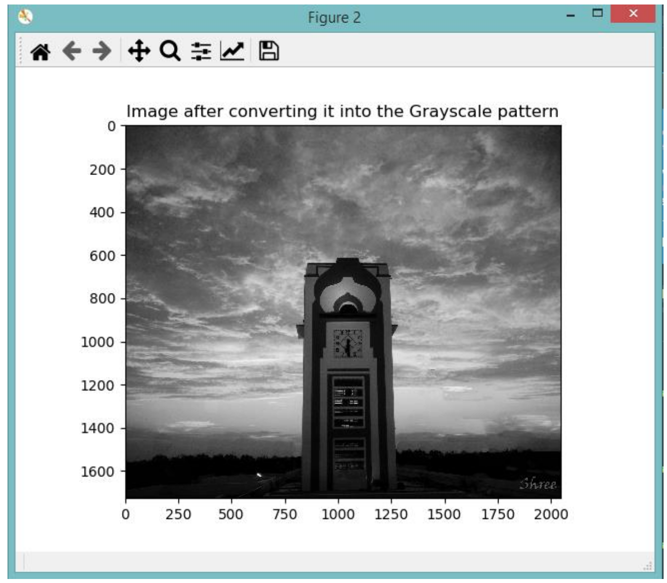
Fig.4 PSNR vs Compression Ratio and SSIM vs Compression Ratio graph plot for 64x64 block SVD.

3.1 PYTHON CODE FOR IMAGE COMPRESSION USING SVD:

```
import os
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
#%matplotlib inline

path=r"C:\Users\Nitish Srivatsa\Desktop\LA Project\test.jpg"
img = Image.open(path)
s = float(os.path.getsize(path))/1000
print("Size(dimension): ",img.size)
plt.title("Original Image (%0.2f Kb):" %s)
plt.imshow(img)

imggray = img.convert('LA')
imgmat = np.array( list(imggray.getdata(band = 0)), float)
imgmat.shape = (imggray.size[1], imggray.size[0])
imgmat = np.matrix(imgmat)
plt.figure()
plt.imshow(imgmat, cmap = 'gray')
plt.title("Image after converting it into the Grayscale pattern")
plt.show()

print("After compression: ")
U, S, Vt = np.linalg.svd(imgmat) #single value decomposition
for i in range(5, 51, 20):
    cmpimg = np.matrix(U[:, :i]) * np.diag(S[:i]) * np.matrix(Vt[:i,:])
```

```
plt.imshow(cmpimg, cmap = 'gray')
title = " Image after =  %s" %i
plt.title(title)
plt.show()
result = Image.fromarray((cmpimg ).astype(np.uint8))
result.save('compressed.jpg')
```





Number of Eigen values , k=25

Number of Eigen values , k=24

3.2 SCILAB CODE FOR IMAGE COMPRESSION USING SVD:

```
function imCompressed = compress(imFullOneChannel, SingularValuesToKeep)
    [U, Sigma, V] = svd(imFullOneChannel);
    SingularValues = diag(Sigma)
    imCompressed = U(:,1:SingularValuesToKeep)*diag(SingularValues(1:SingularValuesToKeep))*V(:, 1:SingularValuesToKeep)'
endfunction


im   = imread('C:\Users\DELL\Downloads\whatsapp\aa1.jpg');
imshow(im);


imFull   = double(im);


imCompressed(:, :, 1)  =  compress(imFull(:, :, 1), 50);
imCompressed(:, :, 2)  =  compress(imFull(:, :, 2), 50);
imCompressed(:, :, 3)  =  compress(imFull(:, :, 3), 50);


imCompressedFinal   = uint8(imCompressed);
imshow(imCompressedFinal);
```

Original Image

Image compressed (Mode – 10)

Image compressed (Mode – 25)

Image compressed (Mode – 50)

Image compressed (Mode – 100)

4. RESULTS AND DISCUSSIONS

We use the machines with following specifications for our experiments:

Operating system - Windows 10

RAM - 8 GB

Software - Jupyter Notebook and Scilab

Processor - Intel i5

We use the copydays original images [7] (without any preprocessing) dataset for calculating performance metrics [8]. For each block-size, we have taken the average of each metric obtained for all the images in the dataset. For each block-size, we obtain the maximum rank from the obtained expression in equation (2). Then, from rank 1 to the maximum rank thus obtained, we calculate all the metrics for each of the images in the dataset. To plot in the graph, we have taken the mean value for each of the metrics we used. The data points in the graphs represent different values for selected rank or k that decrease with increase in compression ratio. Now, we discuss the performance metrics we used for our experiments.

A. Compression Ratio

The compression ratio for an image can be defined as the ratio between the memory size of the original image to that of the compressed image .

$$Compression\ Ratio = \frac{Size\ of\ original\ image}{Size\ of\ compressed\ image}$$

B. Peak Signal to Noise Ratio

Peak Signal to Noise Ratio or PSNR as it is commonly called is an important metric for assessing quality of the compressed image with respect to the original image. As the name suggests it is a ratio between the peak signal value and the noise in the image [9].

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right)$$

Where, R is the maximum signal value 255 in case of gray level image.

C. Structural Similarity Index

Structural Similarity Index often abbreviated as SSIM is a measure for the distortion with respect to the human visual system [10].

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{({\mu_x}^2 + {\mu_y}^2 + c_1)({\sigma_x}^2 + {\sigma_y}^2 + c_2)}$$

Where,

μ_x, μ_y denote the mean value along x and y respectively;

c_1, c_2 denote two variables to stabilize the division when denominator is too small; and

σ_x, σ_x denote the variance along x and y respectively.

Fig. 1, 2, 3 and 4 show the experimental performance of the SVD based compression with the proposed technique and the traditional SVD based image compression. Objective and subjective performance are plotted as compression ratio against PSNR and SSIM respectively. Based upon the results, following observations can be made:

- It is very clear by the results that compression ratio increases by the new technique while PSNR and SSIM remain the same.
- The points in the graphs represent different ranks upon which the metrics are plotted. It is clear from the graphs that the gain in compression ratio is higher as the rank increases while at rank 1, it is zero i.e compression ratio is same for both techniques.
- For the block-size 64, the graph shows decrease in the PSNR and SSIM (as shown in fig. 4), while theoretically there should be no change. The decrease is due to the precision error in calculating the values upon solving the linear equations. The error is negligible upon small ranks but is evident when the rank chosen is larger.

D. Complexity Analysis

The proposed method doesn't create any computational overhead at compression side but however solving the linear equations while decompressing raises the complexity to the order of O($k \times n \times n$) . But since the complexity of SVD itself is of the order of O(n^3), so the overall time complexity of the algorithm remains the same.

5. SUMMARY AND CONCLUSIONS

1) From the above investigation, it is observed that SVD gives good compression results with less computational complexity compared to other compression techniques.

2) A modification to the SVD based image compression is proposed. The modified expressions are derived and by experiments shown are proposed to be better than that of the previous technique.

3) The results can be interpreted to as showing the natural trade-off between selected rank, k (number of Eigen values chosen) and compression ratio but with greater efficiency, while keeping the complexity same.

4) A certain degree of compression as required by an application can be achieved by choosing an appropriate value of k (i.e. the number of eigen values). In other words, degree of compression can be varied by varying the value of k.

5) Smaller the value of k, the more is the compression ratio (i.e. less storage space is required) but image quality deteriorates (i.e. larger MSE and smaller PSNR values).

6) As the value of k increases, image quality improves (i.e. smaller MSE and larger PSNR) but more storage space is required to store the compressed image.

7) Thus, it is necessary to strike a balance between storage space required and image quality for good image compression.

FUTURE SCOPE:

1. Not much work had been done in SVD based image compression, reason being its high complexity. But now with the modification in the expressions, the technique can be further extended to identifying similar blocks and skipping them and using vector coding for the eigenvectors to deliver much more efficient performance.
2. That is, a more effective use of adaptive rank scheme can be achieved by dynamically selecting the block size to be compressed. This can be done by either by repeatedly sub-dividing the original image until an optimal block size is found or by merging smaller blocks together to form a bigger block. Either approaches would require more computation time. The benefit of performing this technique would have to be investigated further.
3. If we could find an efficient technique to obtain the SVD of the bigger blocks simply from the SVD of the smaller blocks, it would have great potential for many applications behind image compression. This would be a very challenging task.

6. BIBLIOGRAPHY

1) Math.mit.edu, Linear Algebra, Chapter 7, The Singular Value Decomposition
2) Linear Algebra and Its Applications, Gilbert Strang, 4th Edition, Chapter 6, Section 6.3, Singular Value Decomposition
3) https://www.researchgate.net/publication/333421316_An_Improved_SVD_based_Image_Compression
4) http://www.ijoart.org/docs/Image-Compression-using-Singular-Value-Decomposition.pdf
5) ***References:***
   - [1] S. Jayaraman, Essaki Rajan, T. Veerakumar, Digital Image Processing, PSG College of Technology, Coimbatore, Tamil Nadu, India, TMH, 2010.
   - [2] Dr.Tinku Acharya and Dr.A. K.Ray , Image Processing principles and applications.
   - [3] Rafael Gonzales, Richard Woods, Digital Image Processing.
   - [4] Singular Value Decomposition Applied To Digital Image Processing by Lijie Cao, Division of Computing Studies, Arizona State university Polytechnic Campus, Arizona.
   - [5] The Singular Value Decomposition and Its Applications on Image Processing by Christopher Jason Ogden, Tammie Huff, Math 45, College of the Redwoods, December 17, 1997.
   - [6] Eldrich, John (2006), "Eigenvalue, eigenfunction, eigenvector, and related terms ", in Jeff Miller (Editor), Earliest Known Usesof Some of the Words of Mathematics, http://jeff560.tripod.com/e.html, retrieved 2006-08-22; Strang, Gilbert (1993), Introduction to linear algebra, Wellesley- Cambridge Press, Wellesley
   - [7] http://lear.inrialpes.fr/people/jegou/data.php
   - [8] Jégou, H., Douze, M., & Schmid, C. (2008). Hamming embedding and weak geometry consistency for large scale image search-extended version.
   - [9] Andrews, H., & Patterson, C. L. I. I. I. (1976). Singular value decomposition (SVD) image coding. IEEE transactions on Communications, 24(4), 425-432.
   - [10] Wang, L. T., Hoover, N. E., Porter, E. H., & Zasio, J. J. (1987, October). SSIM: a software levelized compiled-code simulator. In Proceedings of the 24th ACM/IEEE Design Automation Conference (pp. 2-8). ACM.