

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION**

**PES University**

Outer Ring Rd, Banashankari 3rd Stage,

Bengaluru, Karnataka 560085

2020-2021

Project Report on

**“Image Compression using SVD”**

By

|  |  |  |  |
| --- | --- | --- | --- |
| REG NO | STUDENT NAME | EMAIL ID | CONTACT NO. |
| PES1201802830 | Rohan N Kalpavruksha | rohankalpavruksha@yahoo.com | 6362113556 |
| PES1201800624 | Jayant Saxena | jayantsaxena2000@gmail.com | 7706900470 |

Under the guidance of

Ms.Lavanya Krishna

Faculty, ECE Department

**Image Compression using SVD**

**Executive Summary**

This project is about image compression using SVD. Image compression is a type of data compression applied to digital images to perform image analysis and reducing the cost and time for storage and transmission. But during this process the important features of the image should remain preserved. One such method which serves the purpose is Singular Value Decomposition where images are compressed for different singular values and error between original image and compressed image is calculated.

**Introduction**

It is well known that the images, often used in variety of computer applications, are difficult to store and transmit. For example images captured by satellites are to be transmitted really fast to the space station, storage of medical images in hospital are some of the problems which require to be conquered by image compression.

We can decompose a given image into the three colour channels red, green and blue. To do this, we compute an approximation to the matrix A that takes only a fraction of the space to store. The data in the matrices U, S and V is sorted by how much it contributes to the matrix A in the product, this is based on the fact that the diagonal elements of matrix S are arranged in descending from top to bottom(making it easier to discard irrelevant information from the image) i.e using only the most important parts of the matrices. We now choose a number N of singular values that we are going to use for the approximation. The higher this number, the better the quality of the approximation gets but also the more data is needed to encode it.

**Review of Literature**

1. In 1976, SVD based image coding scheme was proposed by Andrews, H., & Patterson that applied SVD transform for efficiently coding the SVD decomposed image for reducing its size. It was established that the SVD is the most optimal energy compaction model for an image. Based upon the experiments, it was established that due to the some unique properties of the first singular vector, it needs to be coded in a different way to achieve better efficiency.
2. James Chen (ECS 289K Scientific Computation.) in Dec. 13, 2000 published a report based on his work in the field of image compression using SVD. His report consists of several sections that discuss different aspects of the S.V.D. based image compression scheme. First, a straight-forward compression and decompression scheme will be used to show the basic idea of reducing storage requirement with SVD. Second, various block-size will be tested on the same image to compare their effects on the quality of the compression. Third, a rank selection scheme that is adaptive to the complexity of the image has been introduced and discussed.
3. Tian, M, Luo, S. W., & Liao, L. Z. presented a thorough study over the application of SVD for image compression. In the study, it was shown by experiments that the same Eigen vectors cannot be used with different singular values to generate different images as it was shown in some previous studies. Several methods including direct, block based and subtraction of mean before applying SVD were explored for the purpose of image compression. However the results presented could not be verified. The reported PSNR values are very much high as compared to that we found in our experiments for the same image.
4. Rufai, A. M., Anbarjafari, G., & Demirel, H. proposed an image compression scheme that uses SVD followed by WDR (Wavelet Difference Reduction) technique for the purpose of lossy image compression. First, the image approximation is performed using SVD by neglecting small singular values and then using WDR technique, the image is finally compressed by taking wavelet transform and performing bit plane encoding over the coefficients thus generated.
5. Ochoa, H., & Rao, K. R. proposed a modified hybrid DWT- SVD based image compression technique in which the image is divided into blocks and then for each block a decision is taken based upon the standard deviation exhibited by the corresponding pixels. If the standard deviation is high, SVD is used otherwise DWT is used to encode the block. For encoding SVD, Scalar quantization and vector quantization techniques were used. If the MSE of the eigenvector and the coded word is less than a threshold, the index is used otherwise the eigenvector is quantized to 7-bits (first eigenvector) and 5-bits (other eigenvectors).

**Hardware**

Windows XP and above or Mac OS Linux

**Software**

MATLAB

Image Processing Toolbox

**Process:**

**Read and convert image**

inImage=imread('fruits.png') ->Read image

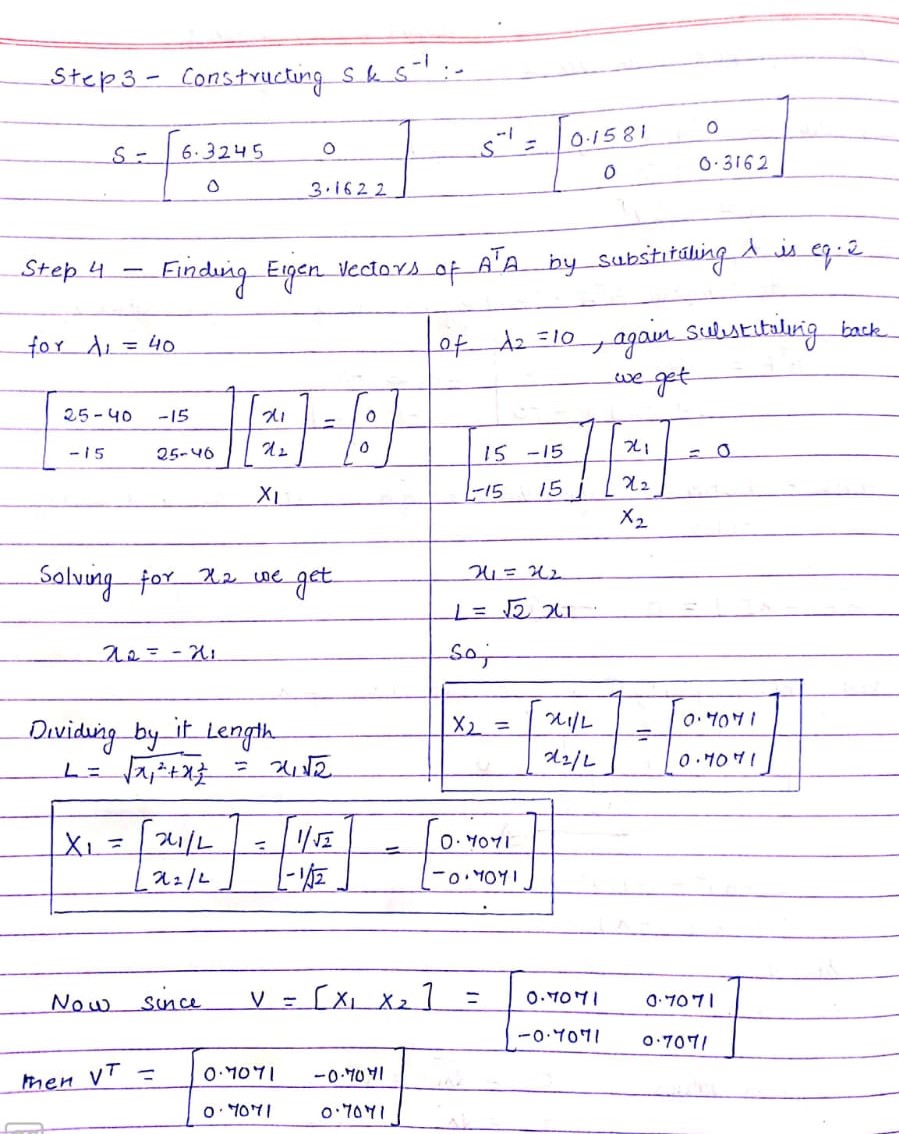
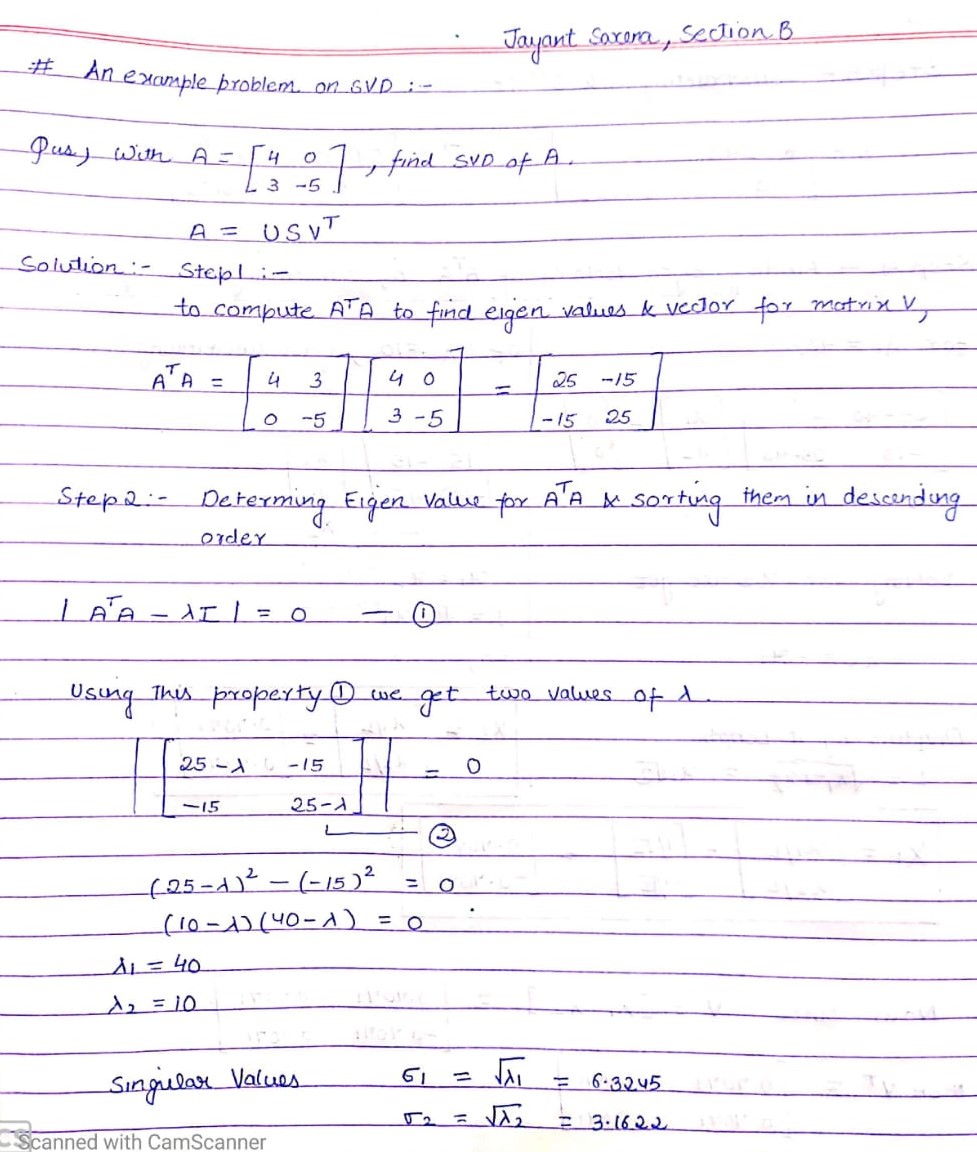
inImage=rgb2gray(inImage); ->Convert RGB to Gray

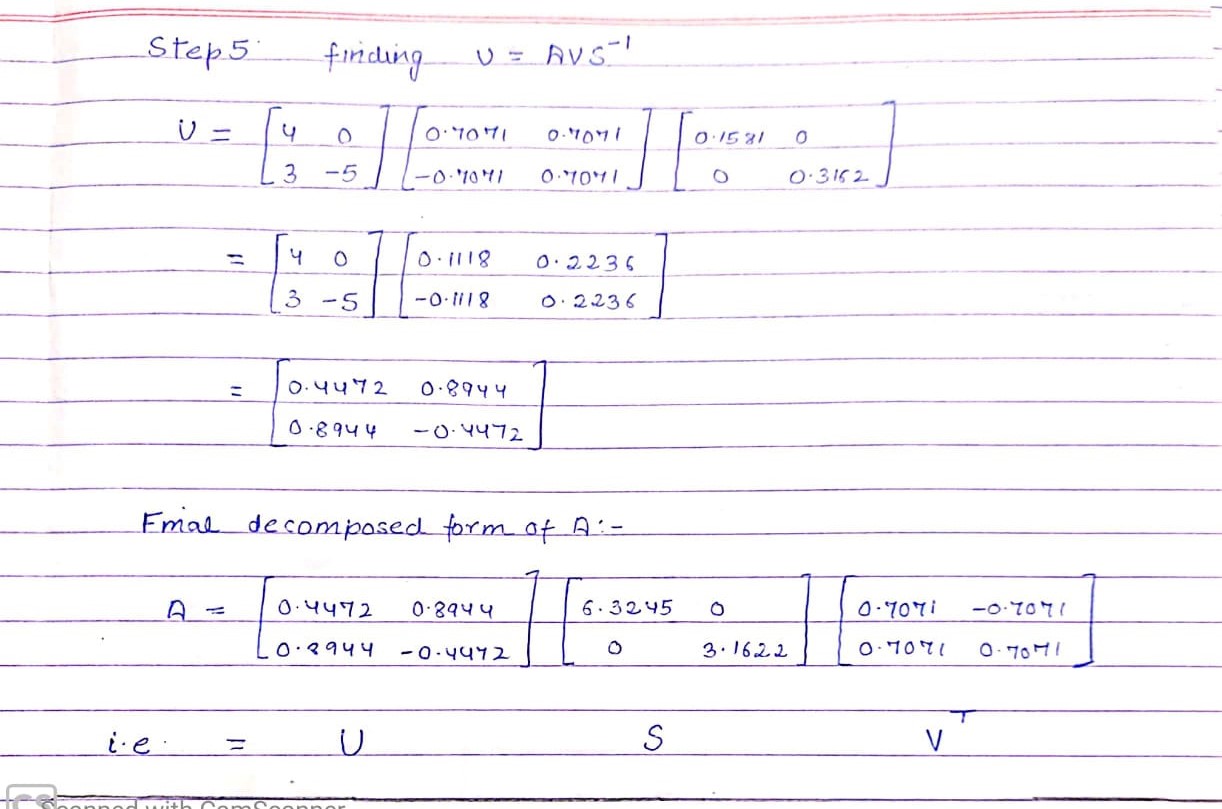
inImageD=double(inImage); ->Rescaling the image from [0-255] to[0-1]



**Decomposing the image using singular value decomposition**

[U,S,V]=svd(inImageD); ->Calculates svd for double image





**Using different number of singular values (diagonal of S) to compress and reconstruct the image**

dispEr = []; ->for storing error values

numSVals = []; ->for storing different singular values

for N=5:25:300

C = S; -> store the singular values in a temporary var

C(N+1:end,:)=0; -> discard the diagonal values not required for compression

C(:,N+1:end)=0;

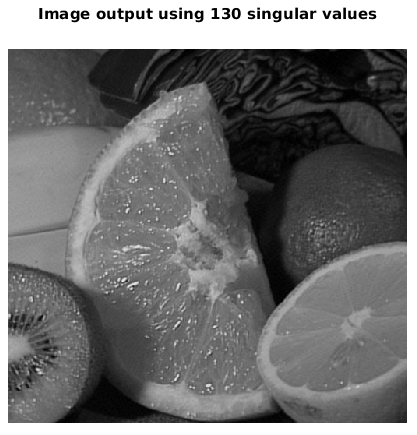
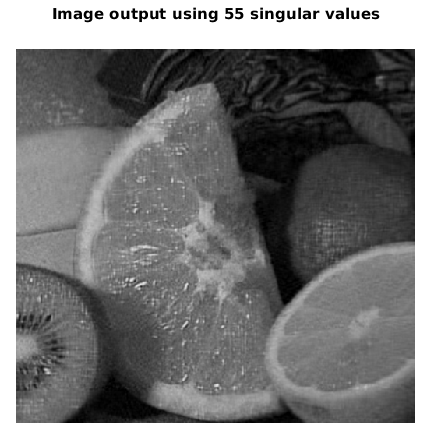
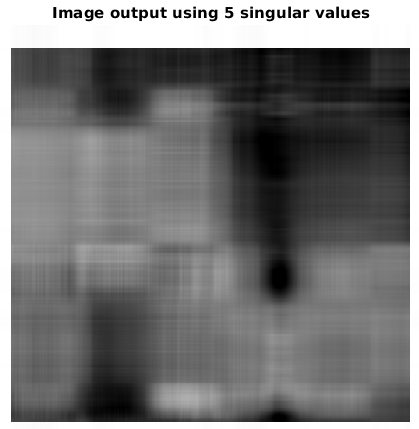
D=U\*C\*V'; -> Construct an Image using the selected singular values

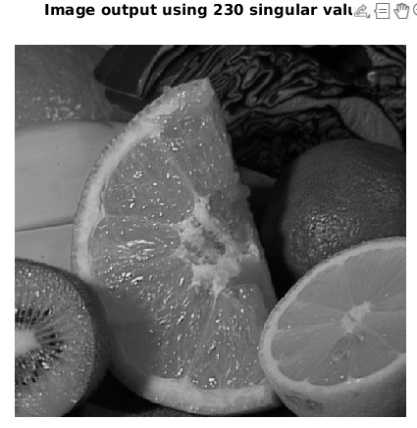
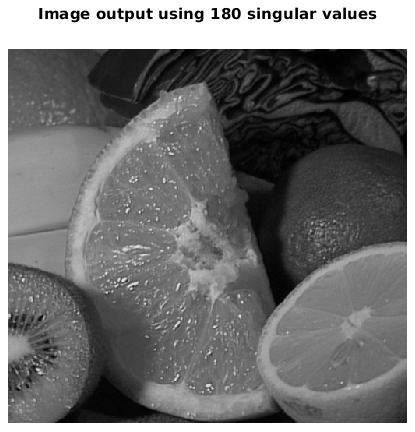
figure;

buffer = sprintf('Image output using %d singular values', N)

imshow(uint8(D));

title(buffer);

****

****

**Compute Error and display its Graph**

error=sum(sum((inImageD-D).^2)); ->Calculation of error

dispEr = [dispEr; error]; -> store vals for display

numSVals = [numSVals; N];

end

figure; -> dislay the error graph

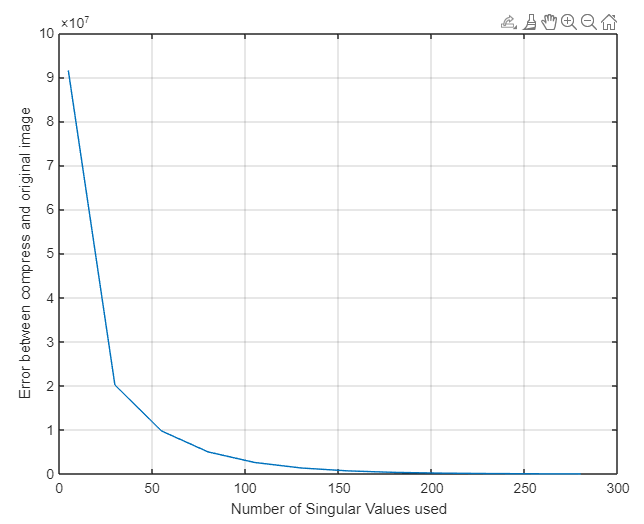
title('Error in compression');

plot(numSVals, dispEr);

grid on

xlabel('Number of Singular Values used');

ylabel('Error between compress and original image');

****

**Results**

Hence report proves that S.V.D. is one of the viable techniques for image compression. We clearly see that with decrease in rank of the matrix(image) the size of the image reduces drastically ,like for our example image (a) the rank(k) of the matrix is 285 which takes up 104310 pixels(i.e around 305.5 Kb) of space in the memory , we reduced the value of k to 20 and saw that the image quality did deteriorate a little but now the images was reduced to 13040pixels(i.e around 38.2Kb) of space.

Hence this technique proves to be a feasible option for image compression.

**References**

(i)<http://elib.mi.sanu.ac.rs/files/journals/ncd/12/ncd12017.pdf>

(ii)J. Chen, « Image compression with SVD », ECS 289K Scientific Computation, 2000,pp.13

(iii)<http://timbaumann.info/svd-image-compression-demo/>