Singular Value Decomposition Application in Image Compression and Face Recognition

[1]Ashwini A. Patil, [2]J. H. Patil, [3]S. P. Patil

[1]Student of M.E., [2]Assistant Professor, [3]Assistant Professor
Department of Electronic and Telecommunication Engineering,
D. N. Patel College of Engineering, Shahada, India

***Abstract*: This paper has applied theory of linear algebra called "singular value decomposition (SVD)" to digital image processing. Two specific areas of digital image processing are investigated and tested. One is digital image compression, and other is face recognition. SVD method can transform matrix A into product USV^T, which allows us to refactoring a digital image in three matrices. The using of singular values of such refactoring allows us to represent the image with a smaller set of values, which can preserve useful features of the original image, but use less storage space in the memory, and achieve the image compression process. The experiments with different singular value are performed, and the compression result was evaluated by compression ratio and quality measurement. To perform face recognition with SVD, we treated the set of known faces as vectors in a subspace, called "face space", spanned by a small group of "base-faces". The projection of a new image onto the base-face is then compared to the set of known faces to identify the face. All tests and experiments are carried out by using MATLAB as computing environment and programming language.**

***Index Terms*: Image processing, Image Compression, Face recognition, Singular value decomposition (SVD).**

I. INTRODUCTION

In linear algebra, the idea of the matrix, including its rank, action on vectors, and geometric structure, are integral aspects of the mathematical study area. Because of this, the Singular Value Decomposition (SVD) is very widely used and incredibly important. The SVD can be used in a huge variety of applications including least square approximations to solving systems of linear equations which all have to something to do with the rank of the matrix. The SVD is very good at approximating matrices of specific rank and can, therefore, be quite applicable in these areas. Specifically, throughout this project, we will be studying the use of the SVD's ability to expose the underlying geometry of an image matrix and make a comparable approximation of the parent matrix.

Compression refers to reducing the quantity of data used to represent a file, image or video content without excessively reducing the quality of the original data. Image compression is the application of data compression on digital images. The main purpose of image compression is to reduce the redundancy and irrelevancy present in the image, so that it can be stored and transferred efficiently. The compressed image is represented by less number of bits compared to original. Hence, the required storage size will be reduced, consequently maximum images can be stored and it can be transferred in faster way to save the time, transmission bandwidth [1].

Face recognition is a necessity of the modern age as the need for identification of individual has increased with the globalization of the world. Personal authentication through face has been under research since last two decades [2]. A facial recognition methodology is a way to automatically verify person by matching his digital image with the database of images. Nowadays the security of person, information or assets is becoming more difficult and important. The crimes like credit card misuse and computer hacking or security breach in organizations are increasing day by day [2]. Facial Recognition is rapidly becoming area of interest [3].

Objective

The objective of this method is to apply linear algebra "Singular Value Decomposition (SVD)" to image processing, especially to area of image compression and recognition. The method is factoring a matrix A into three new matrices U, S, and V, in such way that $A = USV^T$.Where U and V are orthogonal matrices and S is a diagonal matrix.

The experiments are conducted under different term k of singular value, mean square error, compression ratio for image compression; this project also demonstrates how to use SVD approach for image processing in area of Face Recognition (FR).

MATLAB is used as a platform of programming and experiments in this project, since MATLAB is a high performance in integrating computation, visualization and programming.

II. THEORY OF SINGULAR VALUE DECOMPOSITION

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; special feature of SVD is that it can be performed on any real (m, n) matrix. Let's say we have a matrix *A* with *m* rows and *n* columns, with rank *r* and $r \le n \le m$. Then the *A* can be factorized into three matrices:

$$A = U\,S\,V^T \quad (1)$$

Where Matrix U is an m × m orthogonal matrix

$$U = [u_1 \quad u_2 \quad u_3 \quad \dots \quad u_m] \quad (2)$$

And matrix V is an n × n orthogonal matrix

$$V = [v_1 \quad v_2 \quad v_3 \quad \dots \quad v_n] \quad (3)$$

Here, S is an m × n diagonal matrix with singular values (SV) on the diagonal. The matrix S can be showed in following:

$$S = \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_n \end{bmatrix} \qquad (4)$$

III. SVD APPLICATION TO IMAGE PROCESSING

A. SVD APPLICATION FOR IMAGE COMPRESSION

Image compression deals with the problem of reducing the amount of data required to represent a digital image. Compression is achieved by the removal of three basic data redundancies: 1) coding redundancy, which is present when less than optimal; 2) interpixel redundancy, which results from correlations between the pixels; 3) psychovisual redundancies, which is due to data that is ignored by the human visual.

SVD Functions:

Singular value decomposition of symbolic matrix can be easily done using MATLAB build in function 'svd'. This function decomposes the given matrix into three matrices [4]

Syntax:

```
sigma = svd(X)
[U,S,V] = svd(X)
[U,S,V] = svd(X,0)
[U,S,V] = svd(X,'econ')
```

ALGORITHM

Step-1:

Read the image (input image).

syntax:

```
img=imread('filename.jpg');
```

Step-2:

Split the input image (colour image) into R, G, B channels.

Syntax:

```
red = img(:,:,1); % Red channel
green = img(:,:,2); % Green channel
blue = img(:,:,3); % Blue channel
```

Step-3:

Decompose each component using Singular Value Decomposition

Syntax:

```
[u,s,v]=svd(I);
```

Step-4:

Select r value and discard the diagonal value of S matrix not required.

Construct the image using the selected singular values.

Syntax:

```
for j=1:r
    c=c+s(j,j)*u(:,j)*v(:,j).';
end
```

- The r-value in the m-file represents the number of iterations taken on each layer used in the resulting decomposition. This is actually the rank of the SVD matrix. By increasing the rank, we can increase clarity until an optimal image is reached.

Step-5: Display the compressed image.

Evaluation of Compressed Image

To calculate the quality of the image there are different parameters, normally used are Peak Signal to Noise Ratio and Mean Squared Error. In this section, different parameters are discussed [5].

a. Compression Ratio: Compression Ratio is the ratio of the storage space required to store original image to that required to store a compressed image and is given by:

$$\text{Compression Ratio} = m*n / (k*(m+n+1)) \qquad (5)$$

It measures the degree to which an image is compressed.

b. Mean Square Error(MSE): MSE is the measure of deterioration of image quality as compared to the original image when an image is compressed. It is defined as square of the difference between pixel value of original image and the corresponding pixel value of the compressed image averaged over the entire image. Mathematically,

$$\text{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j) - y(i,j))^2 \qquad (6)$$

c. Power Signal to Noise Ratio (PSNR): As the name suggests, Peak Signal to Noise Ratio (PSNR) is the ratio of maximum signal power to the noise power that corrupts it. In Image compression, maximum signal power refers to the original image and noise is introduced to compress it. In other words, noise is the deviation of the compressed image from the original one. Therefore, it follows that PSNR gives the quality of the reconstructed images after compression. Mathematically, PSNR is given by:

$$PSNR = 10\ \log_{10}\left(\frac{255^2}{MSE}\right) \qquad (7)$$

B. SVD Application For Face Recognition

Over the past decades, face image compression, representation and recognition has drawn wide attention from researchers in arrears of computer vision, neural network, pattern recognition, machine learning, and so on. The application of face recognition includes: Access Control based on the face recognition, Computer human interaction, Information Security, Law enforcement, Smart Car etc. SVD approach treats a set of known faces as vectors in a subspace, called "face space", spanned by a small group of "base faces" [6].

In this case; we redefined the matrix A as set of the training face. Assume each face image has m × n = M pixels, and is represented as an M ×1 column vector f_i, a 'training set' S with N number of face images of known individuals forms an M × N matrix.

Algorithm for Face Recognition using SVD

Step 1. Obtain training set S with N face images of known individual.

A training set S with N number of images of known individuals forms an M×N matrix :

$$S = [\, f_1\ ,\ f_2\ ,\ldots\ldots\ldots\ldots,\ f_N\ ] \qquad (8)$$

Step 2. Compute the mean face $\bar{f}$ of S by equation below:

$$\bar{f} = \frac{1}{N}\ \sum_{i=1}^{n}\ f_i \qquad (9)$$

Step 3. Forms a matrix A with the computed $\bar{f}$.

Substracting $\bar{f}$ from the original faces gives

$$a_i = f_i - \bar{f} \qquad \text{where, } i = 1,2,\ldots\ldots\ldots\ldots,N \qquad (10)$$

This gives another M×N matrix A :

$$A = [\, a_1\ ,\ a_2\ ,\ldots\ldots\ldots,\ a_N\ ] \qquad (11)$$

Since = [u_1 , u_2 , … u_r… u_{r+1}…, u_m] form an orthonormal basis for R(A), the range (column) subspace of matrix A. Since matrix A is formed from a training set S with N face images, R(A) is called a 'face subspace' in the 'image space' of m × n pixels, and each u_i

i = 1, 2, ..., r, can be called a 'base face'.

Step 4. Let X (= [x_1 , x_2 , … … …, x_r] ^{T}) be the coordinates(position) of any m × n face image f in the face subspace. Then it is the scalar projection of $f - \bar{f}$ onto the base faces:

$$x = [\, u_1\ ,\ u_2\ ,\ldots\ldots\ldots,\ u_r]^{T}\ (f - f_i) \ldots \qquad (12)$$

This coordinate vector x is used to find which of the training faces best describes the face f. That is to find some training face f_i, i = 1, 2, ..., N, that minimizes the distance:

$$\varepsilon_i = \|x - x_i\|_2 = [(x - x_i)^T (x - x_i)]^{\frac{1}{2}} \qquad (13)$$

Where xi is the coordinate vector of f_i , which is the scalar projection of $f_i - \bar{f}$ onto the base faces:

$$x_i = [\, u_1\ ,\ u_2\ ,\ldots\ldots\ldots,\ u_r]^{T}\ (f_i - \bar{f}) \qquad (14)$$

A face f is classified as face f_i when the minimum ε_i is less than some predefined threshold ε_0 Otherwise the face f is classified as "unknown face". If f is not a face, its distance to the face subspace will be greater than 0. Since the vector projection of $f - \bar{f}$ onto the face space is given

by

$$f_p = [\, u_1\ ,\ u_2\ ,\ldots\ldots\ldots,\ u_r]\ x \qquad (15)$$

where x is given in (12).

The distance of f to the face space is the distance between $f - \bar{f}$ and the projection f_p onto the face space:

$$\varepsilon_i = \left\|(f - \bar{f}) - f_p\right\|_2 = [(f - \bar{f} - f_p)^T (f - \bar{f} - f_p)]^{\frac{1}{2}} \qquad (16)$$

If ε_f is greater than some predefined threshold ε_1, then f is not a face image.

IV. EXPERIMENTS AND RESULTS

A. Results of Image Compression

Figure 1 shows images used for the system tests under different singular values. Figure 2 shows the original image whereas (a) shows the results of the reconstruction image using 10 singular value, b) shows results using 16 values and c) shows the results using 20 values and so on. The observation on those examples, we found when k1 ≤ 20, the images are blurry and with the increase of singular values we have a better approach to the original image


a) Singular value=10


b) Singular value=16


c) Singular value=20


d) Singular value=25


e) Singular value=50


f) Singular value=75


g) Singular value=100

Figure 1 Images with different singular values used for compression

So, in Figure 1 we can see that the image starts to see very decent along the bottom row, the last images using singular values 50, 75, 100.


Figure.2 Original Image


Figure. 3 Compressed Image

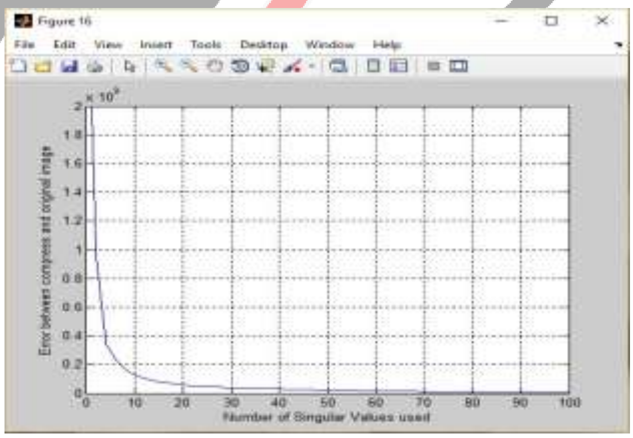

Figure 4. Graph between singular values and error between compressed image and original image

The graph starts to show a heavy turn at about 20 singular values in, and starts to show a decent image, with 50 singular values images is having an extremely low number of error, and image with 75 singular values having a miniscule amount of error. Clearly, from Fig.3 see that image with 100 singular values, and from Fig 4. image is compressed without any loss to image quality, and no blocking artifacts in the compressed image.

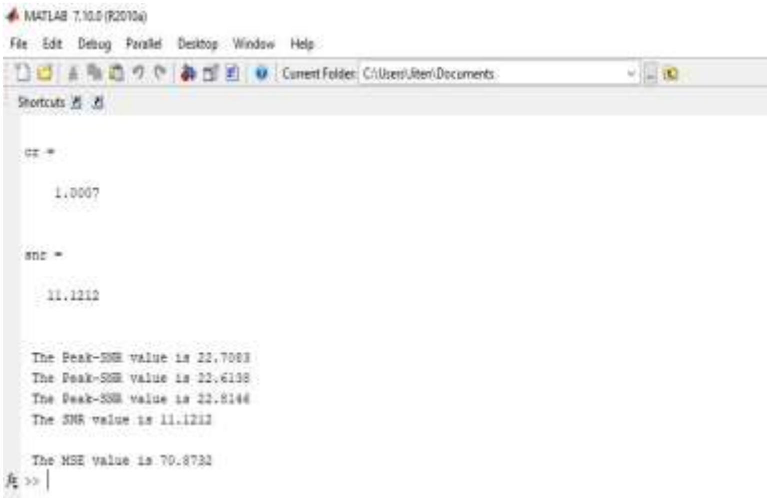

Figure 5. Result of Performance Parameters of image compression

B. Result of Face recognition

The test is under the training set with the number of known individuals: N = 15, Different Conditions: All frontal and slight tilt of the head, different facial expressions. Figure 6 shows the training set of images containing 15 individuals.

Figure 6. Training Set Images

Figure 7. Test Image

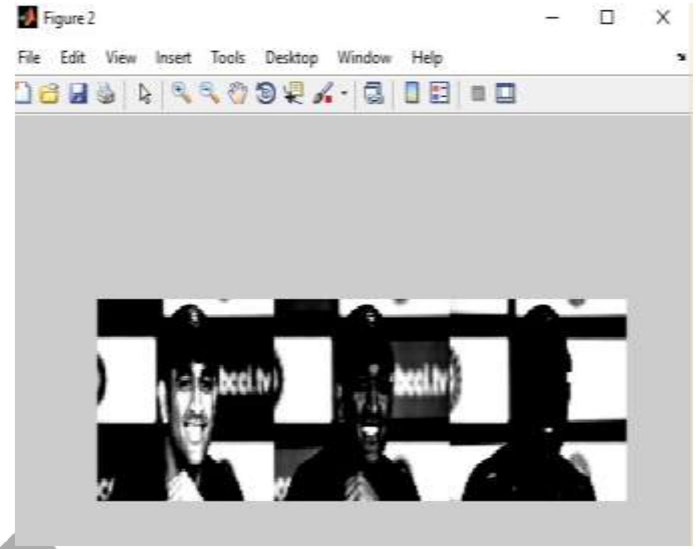

Figure 8. Recognized Image

V. CONCLUSION

This paper has applied technique of linear algebra "singular value decomposition (SVD)" to digital image processing. Two specific areas of image processing are investigated and tested. Basis on the theory and result of experiments, we found that SVD is a stable and effective method. It is observed that SVD gives good compression result without compromising quality of image. During face recognition SVD recognized image successfully, 100% recognition rate is achieved with the help of this proposed method SVD. Basis on the theory and results of experiments, we found that SVD is simple, robust and reliable technique. So, it is concluding that performance of SVD is good for both image compression and face recognition. Overall, the approach is robust, simple, easy and fast to implement. It provides a practical solution to image compression and face recognition.

REFERENCES

[1] Sonal Chawla, Meenakshi Beri, Ritu Mudgil, "Image Compression Techniques: A Review," in IJCSMC, Vol. 3, Issue. 8, August 2014, pg.291 – 296.
[2] Muhammad Naeem, Imran Qureshi and Faisal Azam, "Face Recognition Techniques and Approaches: A Survey." in Sci.Int.(Lahore),27(1),301-305,2015.
[3] B. S. Khade, H.M. Gaikwad, A.S. Aher, K.K. Patil, "Face Recognition Techniques: A Survey," in IJCSMC, Vol. 5, Issue. 11, November 2016, pg.65 – 72.
[4] K. Mounika, D. Sri Navya Lakshmi, K. Alekya, "SVD Based Image Compression," in International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015.
[5] Mr. B Venkataseshaiah, Ms. Roopadevi K N, Stafford Michahial, "Image Compression using Singular Value Decomposition," in International Journal of Advanced Research in Computer and Communication Engineering.
[6] Prashant Jain, Bhupesh Gautam, "Performance Estimation for Face Recognition Using SVD," in International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 1, July 2012.
https://en.wikipedia.org/wiki/Singular_value_decomposition#History
https://lockergnome.com/2006/12/25/real-world-applications-of-image-compression/