

**AMRITA SCHOOL OF ENGINEERING**

**Mathematics for Intelligent System 1**

**21MAT117**

**BRANCH: CSE-AIE**

**SEMESTER-2**

****

**Title: Image processing**

**UNDER THE GUIDANCE OF**: DR. SARADA JAYAN

Batch: E4

TEAM MEMBERS:

M Ramya Sree – BL.EN.U4AIE21072

Poli Vamsi Vardhan Reddy – BL.EN.U4AIE21101

Mettu Siddhartha- BL.EN.U4AIE21080

Abstract

This project has applied theory of linear algebra called “singular value decomposition (SVD)” to digital image processing. Image Compression is the specific area of digital image processing is investigated and tested.SVD method can transform matrix A into product USV’, which allows us to refactoring a digital image in three matrices. The using ofsingular values of such refactoring allows us to represent the image with a smaller set of values, which can preserve useful features of the originalimage, butuse less storage space inthe memory, and achieve the image compression process. The experiments with different singular value are performed, and the compression result was evaluated by compression ratio andquality measurement. To perform face recognition with SVD, we treated the set of known faces as vectors in a subspace, called “facespace”, spanned by a small group of “base faces”. The projection ofa new image onto the base face isthen 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

|  |  |
| --- | --- |
| SI.NO | CONTENTS |
| 1 | Abstract |
| 2 | Introduction |
| 3 | SVD |
| 4 | Properties of SVD |
| 5 | Working |
| 6 | Code |
| 7 | Output |
| 8 | Conclusion |
| 9 | Reference |

INTRODUCTION

Image processing is any form of information processing, in which the input is an image. Image processing studieshow to transform, store, retrieval the image. Digital image processing is the use of computer algorithms to perform image processing on digital images. Many of the techniques of image processing were developed with application to satellite imagery, medical imaging, object recognition, and photo enhancement. With the fast computers and signal processors available in the 2000s, digital image processing has become the most common form of image processing, and is generally used because it is not only the most versatile method, but also the cheapest.

An image can be defined asa two dimension function f (x, y) (2­D image), where x and y are spatial coordinates, and the amplitude off at any pair of (x, y) is gray level of the image at that point.

When x, y and the amplitude value of f are finite, discrete quantities, the image is called “a digital image”. The finite set of digital values is called picture elements or pixels. Typically, the pixels are stored in computer memory as a two-dimensional array or matrix of real number.Colour images are formed bya combination ofindividual 2­D images. Many of the imageprocessing techniques for monochrome imagescan be extend to color image (3­D) by processing the three components imageindividually.

Digital Image Processing (DIP) refers to processing a digital image by mean of a digital computer, and the study of algorithms for theirtransformation. Since the data of digital images in the matrix form, the DIP can utilize a number of mathematical techniques. The essential subject areas are computational linear algebra, integral transforms, statistics and other techniques of numerical analysis. Many DIPalgorithms can be written in term of matrix equation, hence, computational method in linear algebra become an important aspect of the Subject.

Digital Image processing encompassesa wideand variedfield of application, such as area ofimage operation and compression, computer vision, and image analysis (also called image understanding). There isthe consideration ofthree types of computerized processing: low­ level processing is characterized by that both itsinputs and outputs are images; mid­level processing on images is characterized by the fact that its inputs are images, but outputs are attributes extracted from those images, while higher­level processing involves “making sense” of an ensemble of recognized objects asin image analysis, and performing the cognitive function associated with human vision.

SINGULAR VALUE DECOMPOSITION(SVD)

Machine learning extracts information from massive sets of data. The singular value decomposition (SVD) starts with “data” which is a matrix A, and produces “information” which is a factorization A = U ∗ S ∗ V 0 that explains how the matrix transforms vectors to a new space; In many machine learning problems, the massive sets of data can be regarded as a collection of m-vectors, which can be arranged into an m × n matrix. SVD application examples:

1. least squares line, data: points in 2D;
2. curve-fitting, data: polynomial coefficients;
3. matrix approximation, data: entries in a matrix;
4. image compression, data: columns of an image;
5. facial recognition, data: pictures of a face .

Properties of SVD

* There  are  many properties and  attributes of SVD, here  we just present  parts of the  properties that we used in this project. ​

1. The singular value σ1,σ2,.....,σn. are unique, however, the matrices U  and  V are not  unique;  ​

2. Since ATA  = VST SV T, so V diagonalizes ATA  , it follows that the vj s are  the  eigenvector of ATA  .

3. Since   AAT =  USST UT  , so it follows that U diagonalizes   AAT and  that the ui ’s are  the  eigenvectors of  AAT . ​

4. If A has rank of r then vj, vj, …, vr form an  orthonormal basis for range  space of AT , R(AT), and uj, uj, …, ur form an  orthonormal basis for .range space A, R(A). ​

5. The  rank  of matrix A is equal to the  number of its nonzero singular values .​

SIMULATION & CODE

**Method-1**

 %reading and converting the image

inImage=imread("flower.jpg");

inImage=rgb2gray(inImage);

inImageD=double(inImage)

% decomposing the image using singular value decomposition

rank(inImageD)

[U,S,V]=svd(inImageD);

% Using different number of singular values (diagonal of S) to compress and

% reconstruct the image

x = [];

y = [];

for N=5:25:300

% store the singular values in a temporary var

C = S;

% discard the diagonal values not required for compression

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

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

% Construct an Image using the selected singular values

D=U\*C\*V';

% display and compute error

figure;

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

imshow(uint8(D));

title(buffer);

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

% store vals for display

x = [x; error];

y = [y; N];

end

% dislay the error graph

figure;

title('Error in compression');

plot(y, x);

grid on

xlabel('Number of Singular Values used');

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



**Method-2**

inputimage=imread("flower.jpg")

I=double(inputimage);

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

[U1,S1,V1]=svd(R);

[U2,S2,V2]=svd(G);

[U3,S3,V3]=svd(B);

Blocksize= 64;

S1(Blocksize+1:125,Blocksize+1:125)=0;

S2(Blocksize+1:125,Blocksize+1:125)=0;

S3(Blocksize+1:125,Blocksize+1:125)=0;

VT1=V1';

VT2=V2';

VT3=V3';

outputimageR=U1(:,1:Blocksize)\*S1(1:Blocksize,1:Blocksize)\*VT1(1:Blocksize,:);

outputimageG=U2(:,1:Blocksize)\*S2(1:Blocksize,1:Blocksize)\*VT2(1:Blocksize,:);

outputimageB=U3(:,1:Blocksize)\*S3(1:Blocksize,1:Blocksize)\*VT3(1:Blocksize,:);

Reconstructedimage=cat(3,outputimageR,outputimageG,outputimageB);

absolutedifference=I-Reconstructedimage;

PSNR=psnr(Reconstructedimage,I);

UU=U1(:,1:Blocksize);

SS=S1(:,1:Blocksize);

VV=V1(:,1:Blocksize);

structurearrayUU=whos("UU");

structurearraySS=whos("SS");

structurearrayVV=whos("VV");

sizeofo=3\*(structurearrayUU.bytes+structurearraySS.bytes+structurearrayVV.bytes);

structurearrayi=whos('I');

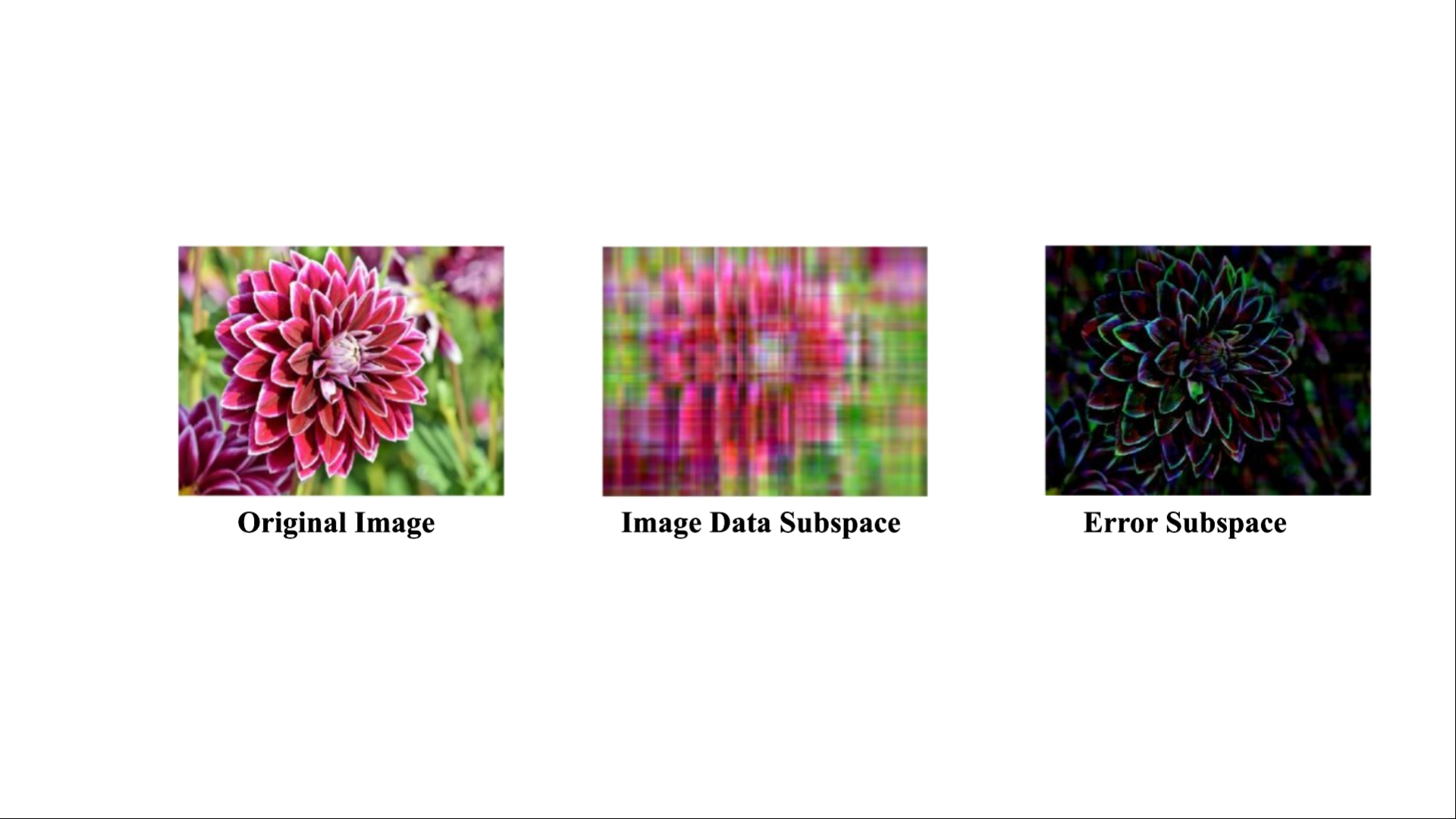
sizeofi=structurearrayi.bytes;

CR=sizeofo/sizeofi;

figure,imshow(I/255)

figure,imshow(Reconstructedimage/255)

figure,imshow(absolutedifference/255)



CONCLUSION

Some machine learning applications involve an M × N array A of data, were N counts the number of data instances, and M measures the size of each data item. The array A can be exactly represented by the full SVD, A = U ∗ D ∗ V 0 Often, the data items have a great deal of similarity and correlation. Then the entries of the D matrix will vary greatly in size. Then the information in A can be well approximated by a low rank approximation using only the first K columns of the SVD factors.

REFERENCE

[1] Rowayda A. Sadek, “SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges” , (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 7, (2012)

[2] M Moonen, P van Dooren, J Vandewalle, “ Singular value decomposition updating algorithm for subspace tracking”, SIAM Journal on Matrix Analysis and Applications (1992)

[3] T. Konda, Y. Nakamura, A new algorithm for singular value decomposition and its parallelization, Parallel Comput. (2009), doi:10.1016/j.parco.2009.02.001

[4] H. C. Andrews and C. L. Patterson, “Singular value decompositions and digital image processing,” IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-24, pp. 26–53, 1976.

[5] Julie L. Kamm, “SVD-Based Methods For Signal And Image Restoration”, PhD Thesis (1998)