IMAGE COMPRESSION USING NEURAL NETWORK

Thesis submitted towards the partial fulfilment of requirement for the award of degree of

MASTER OF ENGINEERING

IN

WIRELESS COMMUNICATION

Submitted By

Sagar Goyal

Roll No. 801363023

Under the guidance of

Dr. Vinay Kumar

Assistant Professor, ECED

Thapar University, Patiala

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC Act, 1956)

PATIALA – 147004, PUNJAB, INDIA

JULY 2015

DECLARATION

I, **Sagar Goyal**, hereby declare that the work, which is being presented in the thesis entitled **"Image Compression using Neural Network"** by me in partial fulfillment of the requirements for the award of degree of Master of Engineering in Wireless Communication from Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Vinay Kumar**, Assistant Professor, Electronics and Communication Engineering Department.

The matter presented in this thesis has not been submitted in any other University/ Institute for the award of any other degree.

Sagar Goyal

Date: 13-07-2015

Sagar Goyal

Roll no: 801363023

This is to certify that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 13-07-2015

Dr. Vinay Kumar

Assistant Professor

ECED

Thapar University, Patiala

Countersigned by:

(Dr. Sanjay Sharma)

Professor and Head, ECED

Thapar University, Patiala

(Dr. S. S. Bhatia)

Dean of Academic Affairs

Thapar University, Patiala

ACKNOWLEDGEMENT

I would like to give special thanks to my guide **Dr. Vinay Kumar**, Assistant Professor, ECED, Thapar University, Patiala, for his advice, kind assistance and invaluable guidance. It has been great honour to work under him.

I am also thankful to **Dr. Sanjay Sharma**, Prof. & Head, Electronics and Communication Engineering Department, for providing us with adequate infrastructure in carrying the work.

I am also thankful to **Dr. Amit Kumar Kohli**, P.G. Coordinator, Electronics and Communication Engineering Department for the motivation and inspiration that triggered me for the report work.

I would like to acknowledge **Dr. Hemdutt Joshi**, Assistant Professor and Program Coordinator (WC), ECED, Thapar University, Patiala, for being a source of inspiration for us during our research work.

I am greatly thankful to all of my friends who constantly encouraged me and also would like to thank all the faculty members of ECED for the full support of my work. I am also thankful to the authors whose work have been consulted and quoted in this work.

Sagar Goyal

Roll No. 801363023

ABSTRACT

In this work, neural network architecture is used for image data compression. It is well suited to the problem of image compression due to their massive parallel and distributed nature. As the digital images require remarkable amount of memory capacity on the disk and also large amount of bandwidth for transmission; therefore image compression algorithms play a significant role in real life applications. One of the successful applications out neural network applications is Principal Component Analysis (PCA) which is used for the image data reduction. PCA is a mathematical technique to transform input data set into lower dimensional space and retain most fundamental data of the original image. In this work, we analyze PCA method with the help of neural network in which free parameters; the synaptic weights act as the principal components which are trained through the iterative method technique known as Generalized Hebbian Algorithm (GHA).

A Comparison with the traditional PCA methods is also performed to demonstrate and illustrate the training and capabilities of Generalized Hebbian Algorithm for image data compression. Simulated data is also presented to evaluate their performance. The evaluated results show that GHA method with neural network architecture gives promising results as the number of iterations increases over both the traditional PCA methods; namely, Singular Value Decomposition (SVD) and Gram Schmidt Orthogonalization Procedure (GSP), in terms of Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Structure Similarity Index (SSIM).

Keywords: *Image Compression, Principal Component Analysis, Singular Value Decomposition, Gram Schmidt Procedure, Generalized Hebbian Algorithm, MSE, PSNR and SSIM.*

Table of Contents

LIST OF ABBREVIATIONS

ART	Adaptive Resonance Theory
ANN	Artificial Neural Network
BP	Backpropagation
CCD	Charged Couple Device
CR	Compression Ratio
DWT	Discrete Wavelet Transform
DPCM	Differential Pulse Code Modulation
DCT	Discrete Cosine Transform
dB	decibel
GHA	Generalized Hebbian Algorithm
GSP	Gram Schmidt Procedure
HVS	Human Visual System
ITU-T	International Telecommunication Union
ISO	International Standard Organization
JPEG	Joint Photograph Expert Group
LZW	Lempel- Ziv-Welch
MSE	Mean Square Error
MLP	Multi-Layer Perceptron
OCR	Optical Character Recognition
PSNR	Peak Signal to Noise Ratio
PCA	Principal Components Analysis
PC	Principal Components
RLE	Run Length Encoding
RBF	Radial Basis Function
SOM	Self-Organizing Maps
SVM	Support Vector Machine
SGA	Stochastic Gradient Ascent
SVD	Singular Value Decomposition
SSIM	Structure Similarity Index
VQ	Vector Quantization

LIST OF FIGURES

LIST OF TABLES

Chapter 1

BASICS OF DIGITAL IMAGE PROCESSING

This chapter gives a brief introduction of the thesis. This includes the concept of digital image processing, basic image compression model, image standards, image formats, and at the end it also discusses about the organization of the thesis.

1.1. Introduction

An image is defined as a two-dimensional function $f(x,y)$, where x and y are the spatial coordinates and the amplitude f at any pair of coordinates (x,y) is called the intensity value of the image at that point [1]. The number of bits used to represent each pixel values are called as pixel depth. Fig. 1.1 represents one of 8×8 block of Lena which is nothing but a two-dimensional array of pixel values ranging from 0 to 255 (assuming that each pixel is using 8-bits to represent an image as shown)

$$\begin{bmatrix} 68 & 85 & 68 & 68 & 68 & 68 & 68 & 68 \\ 68 & 51 & 51 & 68 & 68 & 34 & 34 & 34 \\ 34 & 34 & 51 & 51 & 51 & 34 & 34 & 34 \\ 34 & 34 & 17 & 17 & 34 & 17 & 17 & 17 \\ 17 & 17 & 17 & 34 & 17 & 17 & 34 & 17 \\ 17 & 34 & 17 & 17 & 34 & 17 & 34 & 17 \\ 17 & 17 & 17 & 17 & 34 & 17 & 17 & 17 \\ 34 & 17 & 17 & 17 & 17 & 17 & 34 & 17 \end{bmatrix}_{8x8}$$

Fig. 1.1. 8×8 Block basis of $f(x,y)$ and corresponding pixel matrix

In this matrix, each value 68, 85, 34, 17 and 51 represent individual pixel values of the function $f(x,y)$ at that point. The dimension of the image block is equal to the size of the two-dimensional array.

1.2. Image Formation on Digital Camera

In digital cameras, the image is not formed due to the chemical reaction but it is a little bit more complex operation than analog cameras. In digital cameras [2], Charged couple device arrays of sensors are used for the formation of digital images. Image formation through a CCD array is shown in Fig. 1.2.

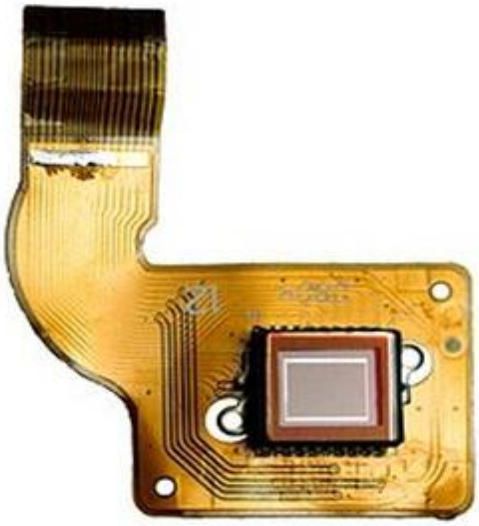


Fig. 1.2. Charged couple device [2]

It is in the shape of a rectangular grid or in the form of array as shown in Fig. 1.3. It is just like a matrix with each cell in the matrix contains a sensor which senses the intensity of the photon. The photon is nothing but just a packet of energy.

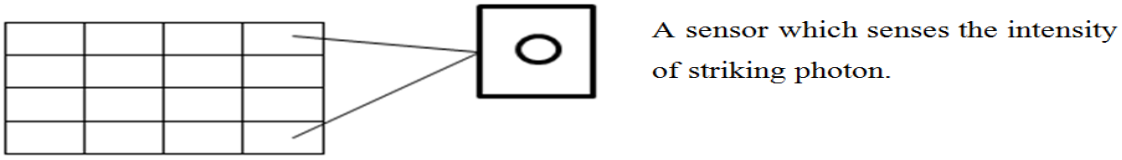
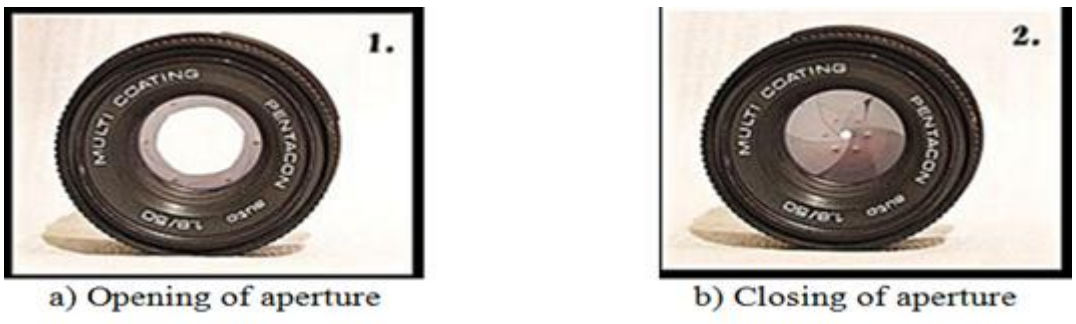


Fig. 1.3. Schematic of charged couple device [2]

Similar to analog cameras, in the digital cameras also, when the light falls on the object it reflects back from the object and enters the light into the cameras through the aperture. Aperture is nothing but just a small opening through which the light enters into the cameras as shown in Fig. 1.4.



a) Opening of aperture

b) Closing of aperture

Fig. 1.4. (a) Opening of the aperture and (b) Closing of the aperture [2]

Fig. 1.4 (b) shows some blades inside the aperture. These blades can be opened or closed. When the more blades are open, the hole through which the light will is bigger and hence the more light is passed into the cameras. The opening and closing of the aperture is directly proportional to the darkness and brightness of an image. When the hole of an aperture is larger then more light is passed into the cameras and hence the brighter image is formed. A complete image is constructed from the two-dimensional array of charged couple device. In digital cameras, there are a less number of sensors and hence limited information is captured. Also in digital cameras, each sensor has only one value against photon which strikes on it. The value of each sensor of the charged couple device refers to the value of the individual pixel i.e., the number of sensors is equivalent to the number of pixels. To measure accurately, external CMOS is attached to the charged couple device array.

1.3. Pixel

The pixel is nothing but smallest element of an image. It is also known as picture element. Each pixel corresponds to only one value. In an 8-bit grayscale image, the range of pixel values is from 0 to 255 where 0 represent the black and 255 represent the white color. The value of each pixel corresponds to the intensity of light, which reflect back from the object and enters into the cameras.

1.4. Pixel Resolution

Resolution is defined as the total number of pixels in a digital image. For example, a digital image has M rows and N columns, then the resolution is defined as $M \times N$. Hence, pixel resolution is defined as a set of two numbers. The first number signifies the breadth of an image, whereas the second number denotes the height of an image. The superiority of an image is proportional to the pixel resolution. It means the higher the pixel resolution, the higher the quality of an image and vice versa.

Generally, two types of digital images are concerned in digital image processing named as grayscale images and color (RGB) images. If we have grayscale images, then each intensity value is represented by 8-bit and the value of intensity lie between 0 and 255 $(2^8 = 256)$ where 0 represent the black color, 255 represent the white color and many shades of a gray in between them. It is also known as a 8-bit image. On the other hand, in RGB images, each of the red, green and blue color platelets are represented by 8-bits.

Hence, 24-bits are used to signify each pixel value. Total number of intensity levels to denote an RGB image are (2^{24}) =16,777,216.

1.5. Image Compression and Need of Image Compression

Image data compression is a technique used for decreasing the redundant data, to signify an image. Digital image of size $2452 \times 1488 \times 3$ requires 10945728 bytes (10.94MB) of memory on the disk. Due to its larger size, it is difficult to upload or download through the internet over the fixed bandwidth. Hence, it is cost inefficient and time consuming. After applying the JPEG compression algorithm [3], the size of the image is reduced to 173KB which requires fewer bytes as compared to an uncompressed image and hence it is convenient to deal with real life. In addition, image compression is also used in scanned documentation, remote sensing, web services and mobile services, etc. Generally, images suffer from the redundancies i.e., various amount of data represents the same information in an image. Redundancy of an image is denoted by R, which is given below

$$R = 1 - \frac{1}{CR} \tag{1.1}$$

where CR is the compression ratio.

For example, if CR = 10, put this value of compression ratio in eq. (1.1) and get the value of redundancy is 0.9 which indicates that 90% of the data is redundant in an image.

There are three different types of redundancies present in images which are explained below

1) Spatial Redundancy

The dependencies of a pixel value upon its neighboring pixel value or in other words the neighborhood pixel values are correlated is termed as spatial redundancy. Transforms like DWT, DCT [4] are used to reduce the spatial redundancy in an image.

2) Psycho-Visual Redundancy

HVS is more perceptual to low frequency as compared to high frequency data of an image. In other words high frequency data are redundant for an image. So many compression techniques [5] focus on reducing the high frequency contents of the image while preserving the low frequency contents.

3) Coding Redundancy

A code is a arrangement of symbols (bits) which is used to represent the information of an image. Each piece of information is characterized by a sequence of bits. The shortest code word represents a high probability of gray levels, whereas the longest code word represents a low probability of gray levels. Data compression is accomplished by allocating fewer bits to high probability gray levels than the less probability gray levels. Many compression techniques in literature are focused to reduce the coding redundancy. There are various techniques available to exploit the coding redundancy named as Huffman coding and Arithmetic coding [1]. For example, 8x8 block of Lena image and corresponding pixel matrix is shown in Fig. 1.1.

The probability of the intensity values is found by following equation

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0,1,2.....L\text{-}1 \tag{1.2}$$

where r_k is intensity value of an image, L is the number of intensity values and n_k is the number of times the k^{th} intensity present in an image.

The number of bits used to characterize each value of r_k is $l(r_k)$, then average number of bits need to represent each pixel value is

$$L_{avg} = \sum_{k=0}^{L\text{-}1} l(r_k)\left(p_r(r_k)\right) \tag{1.3}$$

Hence the total number of bits needs to characterize each pixel value is $M \times N \times L_{avg}$

where M and N are the size of image matrix.

The probability of each intensity value of a 8×8 block is given by eq. (1.2) and thereafter, applies the Huffman coding as shown in Fig. 1.5

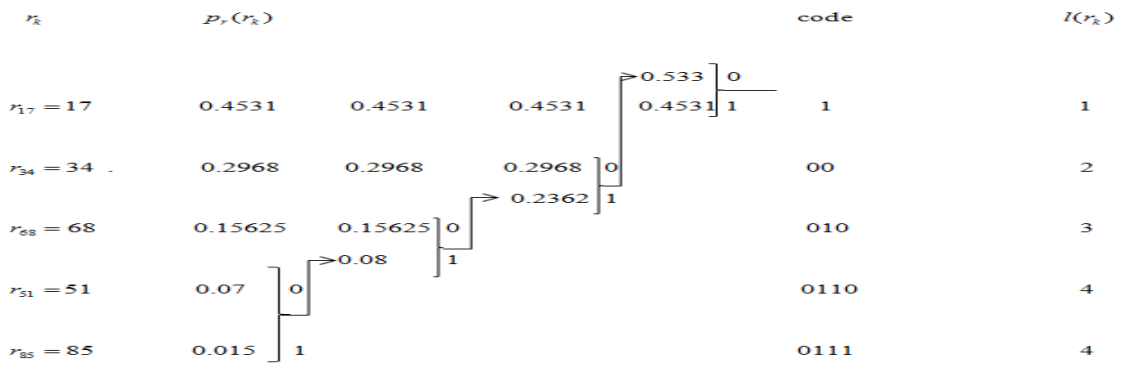


Fig. 1.5. Huffman coding

The average length of encoded pixels is found by eq. (1.3) i.e., $L_{avg} = 1.85$ bits. Total number of bits required to denote 8x8 block of Lena image is $8 \times 8 \times 1.85 = 118$. The resulting compression and relative redundancy is given by

$$CR = \frac{8 \times 8 \times 8}{118} = 4.33$$

and relative redundancy

$$R = 1 - \frac{1}{4.43} = 0.775$$

Hence, 77.5% of the data in 8x8 block of Lena image are redundant.

Image compression algorithms exploited the three defined redundancies named as Spatial Redundancy, Psycho-Visual Redundancy and Coding Redundancy which result in compression of data and reduce the file size of the image.

1.6. Basic Image Compression Model

The image compression model is composed of two well defined functional components, encoder and decoder. The encoder performs compression of data, whereas the decoder performs the complementary operation of encoder i.e. decompression of data. A codec is a device which performs both encoding and decoding. In digital image processing, the input image $f(x, y)$ is served to the encoder, which gives compressed representation of the input image. This compressed representation of image is used for mobile services and web services, etc. When the compressed representation of image is served to the decoder, a reconstructed output image $\hat{f}(x, y)$ is obtained. In an image application, the encoded input and decoder output image is denoted by $f(x, y)$ and $\hat{f}(x, y)$ respectively.

1.6.1. Block diagram of Encoder

Functional block diagram of the encoder is shown below

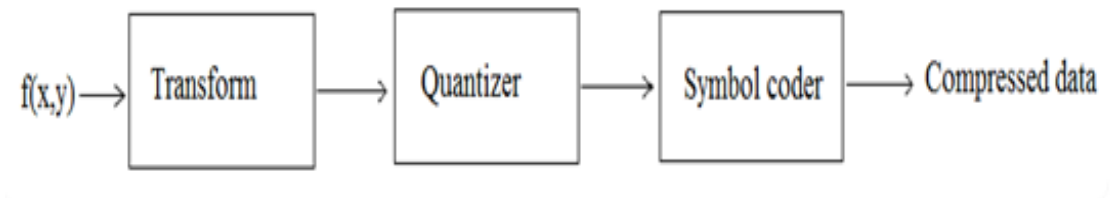


Fig. 1.6. Block diagram of encoder [1]

The encoder is used to remove the redundancies which are present in an image through a following series of operations.

1) Transform (Mapper)

When the original input image $f(x, y)$ is passed through the transforms such as DCT and DWT, it converts the original input image from spatial-domain to frequency-domain. As a result, the low frequency contents lie on the upper-left corner while the high frequency contents are in the lower right corner of the image. As we know that the information on the image changes slowly across the image, especially within a 8×8 block and hence images contain more inter-pixel redundancy. To exploit such type of inter-pixel redundancy, transform, or mappers are used. Various transform techniques in literature [4] are used to exploit the inter-pixel redundancy. Transform block in itself does not perform any compression of data and hence it is a lossless or reversible.

2) Quantizer

It follows the transform block in an image compression system and used a limited number of bits to signify the transformed signal. Here, the quantization matrix $Q(u,v)$ is chosen such that the values of the elements are high at high spatial frequencies and vice-versa. These values lie at the lower-right part of the quantization matrix. In order to quantize the image, the transform coefficient matrix $F(u,v)$ (which is obtained from the output of the transform block) is divided by the quantization matrix $Q(u,v)$ as shown below

$$\hat{f}(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right) \tag{1.4}$$

As a result of the above procedure, the coefficients at the lower right corner of the matrix are reduced to zero. This step is followed Run Length Encoding (RLE) by ZigZag scanning of the matrix. ZigZag scanning [1] is used to create a vector space of data. After creating the vector space, RLE is applied, which replaces the vector space by a pair (Run Length, Value) where Run length is the number of zeros in the run and length is the next non-zero value. ZigZag scanning is shown in Fig. 1.7. A multiple quantization matrix can be used to allow the user how much compression is required. The quality of the image is corrupted as we compressed an image further and further. Hence there is a tradeoff between the quality and compression of the image.

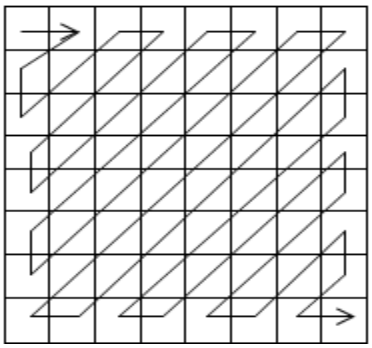


Fig. 1.7. ZigZag scanning [1]

The basic purpose of the quantizer is to exploit the psycho-visual redundancy. As the quantizer perform the compression of data and hence this block is lossy or irreversible. To perform the lossless compression, this block is omitted. For example, 8×8 block of Lena image is shown in Fig. 1. 8. and its 8×8 quantization matrix is shown in eq. (1.5).

Fig. 1.8. 8×8 block basis of $f(x,y)$

$$Q(u,v)=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 29 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}_{8\times8} \tag{1.5}$$

8×8 intensity block $f(x,y)$ of Lena image is shown in Fig. 1.9 and its 2-DCT coefficients are calculated using eq. (1.6) and inverse 2-DCT coefficients are obtained using eq. (1.7). The 2-DCT coefficient matrix is given in eq. (1.8)

DCT II

$$X\left(k_1,k_2\right)=\sum_{n_1=0}^{M-1}\sum_{n_2=0}^{N-1}4x\left(n_1,n_2\right)\cos\left[\frac{\pi}{2M}\left(2n_1+1\right)k_1\right]\cos\left[\frac{\pi}{2N}\left(2n_2+1\right)k_2\right] \qquad (1.6)$$

where $X(k_1,k_2)$ are the 2-DCT coefficients, M and N are the dimensions of an image.

IDCT II

$$x\left(n_1,n_2\right)=\frac{2}{\sqrt{MN}}\sum_{n_1=0}^{M-1}\sum_{n_2=0}^{N-1}X\left(k_1,k_2\right)\cos\left[\frac{\pi}{2M}\left(2n_1+1\right)k_1\right]\cos\left[\frac{\pi}{2N}\left(2n_2+1\right)k_2\right] \qquad (1.7)$$

$$\begin{bmatrix}200&202&189&188&189&175&175&175\\200&203&198&188&189&182&178&175\\203&200&200&195&200&187&185&175\\200&200&200&200&197&187&187&187\\200&205&200&200&195&188&187&175\\200&200&200&200&200&190&187&175\\205&200&199&200&191&187&187&175\\210&200&200&200&188&185&187&186\end{bmatrix}_{8\times8}$$

Fig. 1.9. 8×8 block of $f(x,y)$ and corresponding pixel matrix

$$F\left(u,v\right)=\begin{bmatrix}515&65&-12&4&1&2&-8&5\\-16&3&2&0&0&-11&-2&3\\-12&6&11&-1&3&0&1&-2\\-8&3&-4&2&-2&-3&-5&-2\\0&-2&7&-5&4&0&-1&-4\\3&-3&-1&0&4&1&-1&0\\-2&-2&-3&3&3&-1&-1&3\\0&5&-2&4&-2&2&-3&0\end{bmatrix} \qquad (1.8)$$

In order to quantize the image matrix, 2-DCT coefficient matrix $F(u,v)$ is divided by the quantization matrix. As a result of this, we get the quantized matrix $\hat{f}(u,v)$ whose coefficients is reduced to zero at lower corner of the matrix as shown in eq. (1.9).

$$\hat{f}(u,v) = \begin{bmatrix} 32 & 6 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.9)$$

This step is followed by ZigZag scanning as shown in Fig. 1.8 and it creates a 64-vector space as shown below

(32, 6, -1, -1, 0, -1, 0, 0, 0, -1, 0, 0 ,……..0). After creating the 64-vector space, Run Length Encoding (RLE) is applied, which replaces the vector by a pair (Run Length, Value) where Run length is the number of zeros in the run and length is the next non-zero value as (0,6) (0,-1) (1,-1) (3,-1) (2,1) (0,0).

3) Symbol Coder

It assigns the binary bit stream to the output of the quantizer. It generates [6] fixed or variable bit stream, such as Huffman and Arithmetic coding to minimize the coding redundancy of an image. This block itself does not loss the data of an image so, it is a reversible process.

1.6.2. Block diagram of Decoder

The output of the encoder of an image compression model is in a bit-stream encoded form and hence it cannot be displayed. In an image processing system, the image is sent from one place to another place through the communication channel, which is ideally lossless and lossy for all practical channels. At the receiver end, the compressed image has to be decoded before it has to be displayed. The image decompression system is also known as image decoding system.

Functional block diagram of the decoder is shown below

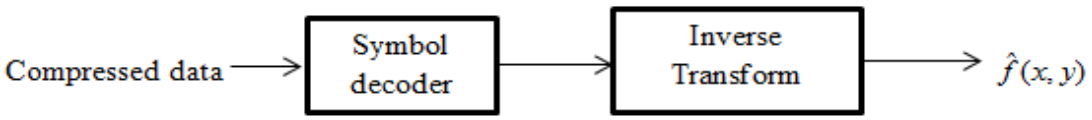


Fig. 1.10. Block diagram of decoder [1]

It contains two components, symbol decoder and an inverse transform. Both symbol decoder and inverse transform perform complementary operations of symbol coder and encoder transform respectively.

1.7. Types of Image Compression

Image compression is generally classified into two categories

1) Lossless image compression.

2) Lossy image compression.

1.7.1. Lossless Image Compression

As the name implies, it compresses the image without incurring any loss of data. It means data before encoding and decoding is exactly the same and no distortion is observed during the reconstruction of the image. For example, Joint Photographic Expert Group-Lossless Compression (JPEG-LS) perform the lossless compression [7].

1.7.2. Lossy Image Compression

Contrary to the lossless image compression, it incurs the loss of data in an image. The techniques under this category involve quantizer which leads to the permanent loss of data which cannot be recovered at the decoding of an image. For example, Joint Photograph Expert group (JPEG) [8] performs the lossy compression. Both the lossless and lossy image compression is shown in Fig. 1.11.

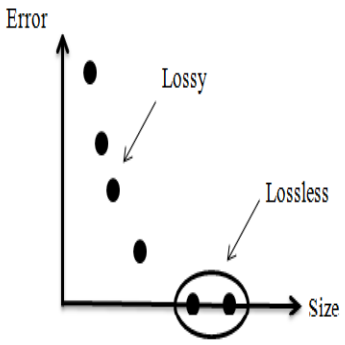


Fig. 1.11. Lossy and lossless image compression[2]

1.8. Image Formats and Image Compression Standards

Image formats define how the image is organized and stored, whereas image compression standards define the procedure of compressing and decompressing of an image to decrease the quantity of data that is essential to characterize an image. These image formats and image compression standards are sanctioned by International Standard

Organization (ISO) and the International Telecommunication Union (ITU-T). Some of the image formats and image compression standard is shown in Fig. 1.12

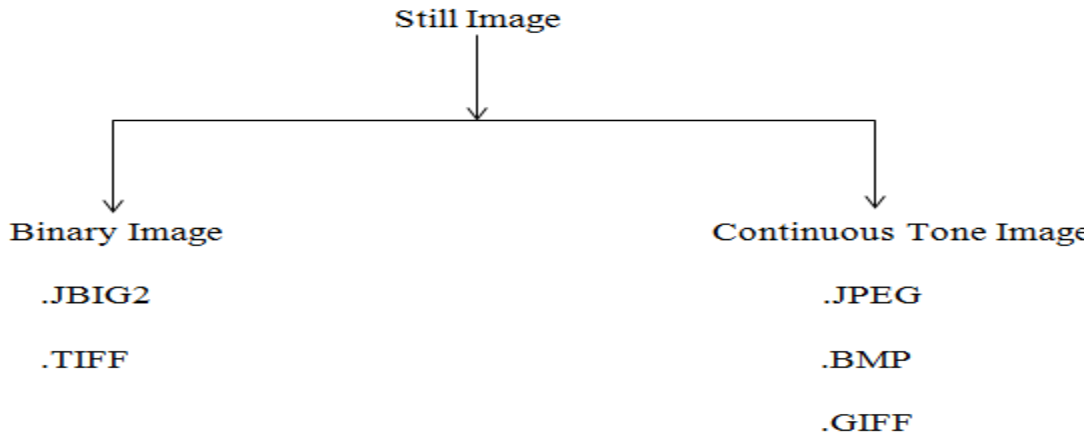


Fig. 1.12. Image formats and image compression standards

1.9. Binary Image

1.9.1. .JBIG2

It is an image compression standard for bi-level images which are computer images in which each pixel value is signified by only one bit. Hence bi-level images contain only two colors, foreground color and background color. Bi-level images are typically created by scanning the images with scanner. JBIG2 is developed by Joint Bi-Level Image Expert Group [9]. It is suitable for both lossy and lossless compression. In JBIG2 standard, scanned image or page is segmented into region of text, region of halftone images and region of other data. Those regions which are not text and halftone, are compressed by using variable length coding technique such as Arithmetic coding.

1.9.2. .TIFF

It is a lossless technique which includes Lempel-Ziv-Welch (LZW) compression algorithm [1] which is considered for the highest quality format for commercial work. In this, there are no additional losses or JPG artifacts to degrade and reduce from the original image. TIFF (Tagged Image File Format) is most dexterous, except for the web pages which do not show the TIFF files. TIFF format is widely used in image applications and page layout applications such as faxing, scanning and Optical Character Recognition (OCR).

1.10. Continuous Tone Image

1.10.1. .JPEG

JPEG stands Joint Photographic Expert Group [8], which is a lossy compression method, it uses quantized Discrete Cosine Transform (DCT) coefficients on 8x8 image blocks, Arithmetic coding, Huffman coding and Run-length coding. In this method, the images are saved with the JFIF (JPEG File Interchange Format) file format. The filename extension of JPEG/JFIF is .JPG or .JPEG. Mostly, all digital cameras store the images in JPEG/JFIF file format, which supports 8-bits of grayscale images and 24-bits for color images, in which eight bits are used for each red, green and blue platelets. As a result, JPEG is lossy compression techniques which affects the visual quality of an image and reduce the file size of an image. It suffers the performance degradation when continuously edited and saved. Generally, the web pages use JPG file format to compress the data and hence occupy the less space on the memory disk.

1.10.2. .BMP

BMP file format is also known as a bitmap image file or simply called as a bitmap. BMP are basically uncompressed files. Their basic advantage is to handle the graphic image file format, especially for the Microsoft windows and for the operating systems.

1.10.3. .GIFF

GIFF stands Graphics Interchange File Format which was designed by CompuServe to replace the Run-Length Encoding (RLE) format. It is limited to 256 colors which make the GIFF format more suitable for storing graphics with very few colors such as simple diagrams, logos and shapes. It uses the Lempel-Ziv-Welch (LZW) lossless compression algorithm for those areas which have large single color in the images.

Now we will discuss about the motivation and thesis objective of my works as follows

1.11. Motivation

Digital Image Processing aims to restore, increase and manipulate information contained in digital images for different purposes by means of digital computers. Computer images require large amount of memory on the disk. For example, a color image of size $2452 \times 1488 \times 3$, requires 10.94 MB of memory on the disk. Due to its large file size, it is difficult to upload or transmit through the internet over the fixed bandwidth. Therefore, it is cost inefficient and time consuming for real life applications. So, there is the need of

the image compression algorithms for the efficient use of the memory on the disk and bandwidth available for the information interchange.

1.12. Thesis Objective

The aim of research in this thesis is to examine and analyze the Principal Component Techniques (PCA) based on neural network architecture for the image data compression, and the use of Generalized Hebbian Algorithm (GHA) with the neural network for image compression, exploiting the important features of an image.

The specific objectives are

1) Discuss the traditional PCA techniques such as Singular Value Decomposition (SVD), Gram Schmidt Procedure (GSP) which are developed for image data compression.

2) A comparison with traditional methods is accomplished to validate and explain the training and capabilities of GHA for the image compression.

1.13. Thesis Organization

The remaining chapters represent the following the research work

- **Chapter 2: Literature Survey**: This chapter provides the information about the literature survey for artificial neural network for image compression.
- **Chapter 3: Basics of Artificial Neural Network**: This chapter gives a brief introduction of the Artificial Neural Network (ANN). This also includes how the ANN is inspired from the biological nervous system, advantages and applications of the artificial neural network, types of training methods and also defines two image compression techniques with the help of artificial neural network.
- **Chapter 4: Methodology and Analysis**: This chapter includes traditional PCA methods, named as Singular Value Decomposition (SVD), Gram Schmidt Procedure (GSP) and the neural network based PCA method called Generalized Hebbian Algorithm (GHA).
- **Chapter 5**: **Results**: In this chapter, we provide results between the traditional PCA methods and neural based method named as Generalized Hebbian Algorithm (GHA).
- **Chapter 6: Concluding Remarks and Future Scope**: At last, in this chapter, we provide concluding remarks and future scope of my research work.

Chapter 2

LITERATURE SURVEY

This chapter provides a literature survey in the field of the Digital Image Processing (DIP) for the image compression technique with neural network available in the literature work.

Jain [10] studied a large number of data, image compression methods such as Differential Pulse Code Modulation (DPCM), Transform coding, Predicting coding, adaptive techniques and its application. Results of channel inaccuracies and other diverse problems are also discussed. Numerous of the examples focus on the visual images which can be easily adapted for multi-dimensional image compression. The focus is on the fundamental techniques used for the data image compression.

Sanger [11] presented a new method for unsupervised learning in a single layer feed forward artificial neural network. An optimal principal was proposed which was based upon maintaining the maximum information on the output neurons. The proposed algorithm centered upon the Hebbian learning mechanism which accomplished the desired response of the network. The algorithm abstracts the principal components of the input covariance matrix and it is shown that it converges with probability one. It is shown that the algorithm is related to the statistics (Principal Component Analysis) and the neural networks (Back-propagation or the encoder problem). It also delivers an explanation of the artificial neural network in terms of the classical techniques.

Oja [12] proposed the numerical method for the Principal Component Analysis (PCA). The connection between adaptive algorithms and the Hebbian learning rules are also studied. The Stochastic Gradient Ascent (SGA) is presented and shown that it is very closely related to the Generalized Hebbian algorithm (GHA). The proposed SGA showed better for removing the less dominant principal components. The stochastic gradient ascent algorithm is extended further for the minor principal components. Two extensions were proposed; in the first one each neuron has some scalar limit which breaks the symmetry of the network. In the second, a random number of parallel neurons are considered not necessary that it is less than the input vector dimension.

Abbas [13] *et al*. proposed an artificial neural network system to execute the adaptive calculation of the principal components of the input covariance matrix. The proposed

system is based on the modified Hebbian learning mechanism which is proposed by the Oja on every new covariance matrix that results after calculating the principal components. The approach is shown that the convergence of the next dominant principal component is linearly free from the previous principal components. The precise learning rate is determined by reducing the error function along the steepest descent direction. The proposed method is applied to the grayscale images, calculating the limiting number of the KLT transform coefficients that meet a specific performance criterion. The altering of the input matrix, the bit rate values, peak signal to noise ratio, compression ratio and generalization ability of the neural network architecture are investigated.

Abbas [14] *et al*. presented an algorithm to increase the speed of the back-propagation learning mechanism. The algorithm gives the lateral connections among the neurons on every hidden layer. These connections between the neurons are trained using an Anti-Hebbian learning mechanism which de-correlates the output of these neurons. The de-correlation process lessens the redundant data which are transferred from an input layer to the next layer and hence increased numerical performance of the neural network.

Dony [15] presented the outline of the neural network for the image compression. The neural network is well suitable to solve the problem of the image compression because of their massively parallel distributed network. The features of the neural network are much similar to the biological nervous system. Here, the multilayer perceptron is used as a nonlinear predictor in the differential pulse code modulation (DPCM). Such nonlinear predictors have presented the high predictive gain as compared to the linear predictors. The application of the Hebbian learning rule is also shown for the removal of the principal components of the correlation matrix. These algorithms are iterative in nature and have the computational benefits over the eigen value decomposition technique. Another model self-organizing feature map (SOFM) has been shown a excessive achievement in the making of the codebooks for the vector quantization (VQ). The proposed codebooks are less sensitive to traditional algorithms and exploited to further increase the coding proficiency and decrease the computational difficulty.

Chin [16] *et al*. studied the data compression in the neural network. Neural network accepts the huge amount of data, such as imagery or text data, compressed it for packing or transmission and frequently restores it when we desired. A new training technique called as the Nested Training Algorithm (NTA) was proposed for reducing the training

time so that the neural network converges quickly. The performance of the neural network is calculated using reconstructed and the real world data. The performance of the proposed design and training algorithm gives high compression ratio and low distortion rate.

Fahmy [17] *et al* presented the new neural network for the data image compression in an image. The suggested technique extended the two layer feed-forward neural network to the multi-layer network. The experiment consequence presented the high performance of the multi-layer network as related to the two-layer feed-forward network particularly at a high compression ratio. To decrease the extensive training period for the multi-layer networks, a new training algorithm was used. A modified feedback error was suggested to reduce the further training time and enhanced the quality of the image. Also, the new training phase was proposed to decrease the mean squared error (MSE) which was more related to the Human Visual System (HVS).

Clausen [18] *et al*. studied the hybrid neural network model to reduce the redundant data in the color image which was based on the Sanger's algorithm to characterize an image in terms of the principal component analysis and the back-propagation algorithm for returning the original representation. The principal component analysis produced a black and white image which has the equal number of pixels as to the original color image, but with each pixel value was denoted by the scalar value instead of three- dimensional RGB components. The proposed hybrid learning method assumed the spatial image features as it outclassed the YUV and YIQ standard compression methods. It was a shown that it was possible to apply the training results of one image to another unseen image.

Costa [19] *et al*. Principal Component Analysis (PCA) is a numerical procedure to de-correlate the redundant data in an image by projecting the original data onto the principal components. The principal component analysis performed both in recursive and batch mode. The former method was one of the very real techniques for high dimensional data, such as in the image compression. The paper presented the comparison between the principal component analysis (PCA) neural networks for still image and the coding. The paper showed the basic concepts of the neural networks PCA and after, it characterized the comparison between the structure and the learning mechanism in the neural networks. The paper also discussed the advantages and disadvantages of each related techniques.

The inference of comparison among the eight principal components is that the cascade recursive least-square algorithm shows the best results and the statistical properties.

Ma [20] *et al*. studied an presentation of an adaptive constructive one-hidden layer feed forward neural network (OHL-FNN) for an image compression. The offered adaptive constructive neural networks were compared with the fixed structure of the neural networks. The comparisons between the quantization effects and with baseline JPEG were also considered.

Li [21] *et al*. Vector quantization (VQ) was an vital technique for an image compression. To obtain the high compression ratio and to hustle up the process, recommended a new technique called as the transformed vector quantization (TVQ) which joint the features of transform coding and the vector quantization. The comparison of reconstructed image quality was made between the vector quantization (VQ) and the suggested transformed vector quantization (TVQ). Also the contrast was made between the standard JPEG and the TVQ approach.

Handels [22] *et al*. discussed the role of artificial neural network, particularly in feed-forward network, Hopfield neural networks and Kohonen feature maps. The numerous applications of neural network are divided into the two-dimensional image processing algorithm. One dimension defines the type of task which is performed by the algorithm: preprocessing, segmentation, image understanding, object recognition, and optimization. The second dimension of the neural network captures the abstraction level of the input data which is treated by the algorithm: pixel-level, structure level, local feature level, and the scene characterization. Each of the tasks has some limitations on the neural networks. A synthesis is made for the uncertain problems related to the pattern recognition techniques in image processing. The future application of the neural network and other developments are also presented.

Fang [23] *et al*. proposed the image compression algorithm which is centered on the PCA or SOFM hybrid neural networks, which have the extreme rewards over both the PCA and SOFM. An offered method of selecting the initial codebook is denoted to progress the proficiency of the SOFM neural network, according to a numerical feature of PCA transformation coefficient. It showed many great advantages such as lower memory storage, computational faster and better performance of the codebook.

Robinson [24] *et al*. presented the algorithm for the application of SVM to an image compression. The anticipated algorithm combined with Discrete Cosine Transform (DCT) which is different from the multilayer perceptron and radial basis function (RBF) networks which necessary the network topology before the training, whereas SVM selected the minimum number of the training points called as support vectors, which ensured the modeling of the data to the given level of accuracy. The support vector machines algorithm performed the image compression in a domain of DCT coefficients. The restrictions of the support vector machines were stored to reconstruct the image. The new proposed algorithm showed that the compression ratio of an image was higher as compared to the other traditional compression techniques.

Leung [25] studied the two global techniques one is PCA and the other is independent component analysis (ICA) which were used to compress the gigantic amount of data in an image by misusing their correlation properties. Both the methods approximated the raw data with a small number of global base images and followed a similar algorithm structure: raw data representation and the base image extraction. The only difference was that PCA removed the second- order data correlation whereas the ICA reduced the all order statistical data dependence, which was a benefit to the data compression. The experimental results of the proposed techniques were more to JPEG and MPEG. Although independent component analysis removed all the higher order statistics than principal component analysis (PCA), it was little inferior to PCA in terms of the compression ratio of an image.

CHAN [26] *et al*. studied various image recognition algorithms which are centered on the dimensionality reduction former to the actual learning because the actual or real data take the massive amount of the training sets to achieve the desired performance. The major problem of most widely used dimensionality reduction technique, like PCA is to maximize representation of data variability into the smallest number of the principal components, without including the interclass discriminability. He offered artificial neural networks to decrease the dimensionality of the data with a high degree of discriminability by combining with the PCA and the back propagation learning rules.

Base [27] *et al*. studied a performance which is based on the topology-preserving neural networks are used to implement VQ for the image compression. The proposed technique is advanced for the image compression, which distinct itself from other traditional

techniques in various ways. This method is practical to non-overlapping image blocks and signifies a better probability distribution method. The quantization process is performed by the neural network, which practices the vector quantization to converge the training time period rapidly and reaches the mean square error lower than that of the Kohnen's feature maps or LBG algorithm.

Soliman [28] *et al*. proposed an artificial neural network model for the image compression, which is called the DC model. The DC model is a hybrid between the subset of SOK and ART model. The DC model is computationally fast for neural network classification. The trained DC model uses the precision of the winners take features of the SOK model and the speed of the ART model. The DC model gave better results than that of the earlier image compression techniques such as JPEG2000 and wavelet technology, chiefly for the colored and the satellite images.

Vilovic [29] presented the direct solution of the image compression using the Multilayer Perceptron (MLP) neural network in digital image processing. It developed the transform coding of the image. The MLP network is trained with dissimilar number of hidden neurons in the hidden layer with direct relation to the compression ratio. The image is divided into different non-overlapping blocks sizes for the compression process. And the reconstructed and the original images are compared with the help of the image quality parameters, i.e. peak signal to noise ratio and the number of bits per pixel.

Northan [30] *et al*. proposed a multiresolution neural network (MRNN) filter bank has been made within a state of the art sub band coding framework. The filter bank synthesizes the signal precisely from the reference coefficients. Hence, the low pass filter banks of the MRNN are trained and recreate the signal accurately. The high pass filter banks of the MRNN are trained for the perfect reconstruction of the image. The MRNN filter banks are used in place of linear filter banks and partitioning the hierarchical trees (SPIHT) coder. The proposed filter banks show the advantages over the linear filter banks such as lower bit rates and better compression ratio.

Durai [31] *et al*. feed forward neural network practices the back propagation algorithm accepting the gradient descent method for the minimization of the error and hence it is directly applied to the image compression. In various literature works were done to achieve the rapid convergence of the neural network without the loss of quality of reconstructed images. Images are compressed with distinct types such as high intensity

image and dark image, etc. When these images are compressed with back propagation algorithm, it will take a large amount of time to converge. The reason for this is that the given image comprises the dissimilar number of gray level pixels which are correlated to their neighborhood pixels. If the gray levels pixels of an image and their neighborhood pixels are plotted in such a way that the difference between the neighboring pixels and the pixels of an image are decreased then the convergence rate and the compression ratio of an image may be improved. To obtain this, a proposed cumulative distribution function is recycled and maps to the image pixels. When the mapped pixels are used in back propagation neural network, it produces the high compression ratio and converges the neural network quickly.

Koren [32] *et al*. developed an algorithm for the image compression and artificial neural networks for ideal positioning of the satellite images. The heart of the suggested algorithm is the program of NNIC-neural network image compressor. The code was settled for grayscale and color image compression with the help of artificial neural network (ANN). Neural network image compressor applied three different techniques for an image compression. Out of three, two of them are based on the artificial neural network structure-multilayer perceptron and Kohonen network. The third is centered on the method of DCT for the JPEG standard. The code also defines the numerical and the visual quality parameters of the compression and comparison between different traditional techniques.

Arunapriya [33] *et al*. proposed the real technique called the Wavelet-Modified Single Layer Linear Feed Forward Only Counter Propagation Network (MSLLFOCPN) for an image compression. This technique accepted the properties of the spatial and the frequency coefficients from wavelets. Prediction and the function approximation were obtained from the neural networks. As a result, counter propagation network was measured superior performance and hence proposed a new neural network called as the single layer linear counter propagation network (SLLC). Several test images were applied to the proposed technique combined of wavelet and the SLLC network. The results of the proposed technique with existing traditional neural network technique showed the compression ratio, prediction and the picture quality were highly superior.

Sahami [34] *et al*. presented the application of the neural networks for the bi- level image compression. In the future image compression algorithm, pixels of an image were

practical as inputs to the Multilayer Perceptron (MLP) neural networks. The outputs of the neural networks were signified by the pixel value 0 or 1. The final synaptic weights of the trained neural networks were quantized and denoted by a few number of bits, Huffman encoded and then stored as the compressed image. While the decompression side, the pixel locations were applied to the trained neural network and the output of the networks determined the intensity value. In the proposed structure, more than 4000 images were verified and indicated the higher compression rate as compared to the other traditional compression techniques such as Comite Consultatif International Telephonique of Telegraphique (CCITT) and the joint bi-level image expert group (JBIG2) standards. Moreover, quantization issue in the neural networks was addressed and the explanation was also proposed. Further, adaptive techniques on the binary images were applied to attain higher compression ratio.

Boopathi [35] *et al*. image compression is a technique which is used for reducing the file size of the image without affecting the quality of the image. Image compression is done so that the more images are stored on the memory disk. It also decreases the time to upload or download the image from the internet. There are several traditional techniques are available for the image compression out of which vector quantization is the most powerful technique for the image compression. Vector quantization is nothing but the generation of the codebook. In the suggested technique, Radial Basis function (RBF) is used to generate the codebook. A combined approach for an image compression based on the vector quantization and the wavelet transform proposed RBF neural networks. The proposed technique is helpful for criminal examination and the medical imaging where the high accuracy of the reconstructed image is mandatory. The proposed technique shows the better PSNR and reduces the mean square error value (MSE).

Amar [36] *et al*. proposed an image validation algorithm in the discrete cosine transform (DCT) domain which was based on the neural networks. It involved the average value of each 8x8 blocks of an image. In addition, neural network was trained and later was used to recover the damaged region of an image.

Chapter 3

BASICS OF ARTIFICIAL NEURAL NETWORK

This chapter gives a brief introduction of the artificial neural network. This chapter also includes how the ANN is inspired from the biological nervous system, advantages of the ANN, types of training methods, types of learning protocols, and also defines two image compression techniques with the help of artificial neural network.

3.1. Artificial Neural Network (ANN)

Artificial Neural Network [37] is a massive and non-linear parallel distributed network, which is interconnected with large processing units known as neurons. It is called as massively and non-linear parallel network because each neuron in the network process independently. Artificial Neural Network is stimulated from biological nervous system. It is made up of a large number of processing units to solve a particular problem such as pattern classification, clustering of the data, and pattern recognition etc. It is just like people who learn through example or experiences. In the biological nervous system, learning is done through the adjustment of the synapses that exist in its processing units. Like to biological nervous system, the same learning is also done in ANN. It is similar to the biological nervous system in two ways.

1) ANN acquires the knowledge through the learning process.

2) Inter-neuron connection which is known as synaptic weights are used to store the knowledge of the network.

3.2. Advantages of Artificial Neural Network

Nonlinearity: Distribution of the neurons in the network is nonlinear which is massively parallel connected to perform a particular task.

Adaptive learning: The adaption of the free parameters such as synaptic weights and basis to perform a particular task through the learning process.

Real time operation: Artificial Neural Network computations may be carried out in parallel, and hardware devices are being designed and manufactured which take advantage of this capability.

Fault Tolerance: Partial destruction of the network leads to graceful performance degradation. It means some network capabilities may be retained even after the major network destruction has happened.

3.3. Biological Nervous System

The processing unit in the biological nervous system is neurons. It contains billions number of neurons which are non-linear and parallel connected in a brain. All natural neurons consist of four basic components which are dendrites, soma, axon and synapses. Generally, the biological nervous system receives the input from one source and performs the non- linear operation on it and hence gives the final output. The figure of biological nervous system and its basic four components is shown below. In the biological nervous system, a neuron collects information from other neurons through fine dendrites. The neurons send out the spikes of electric signal through a long, thin stand known as axon, which further splits into thousands of branches. At the end of each branch, a structure known as synapse is used which converts the activity from the axon into electric effect that inhibit or excite the connected neurons.

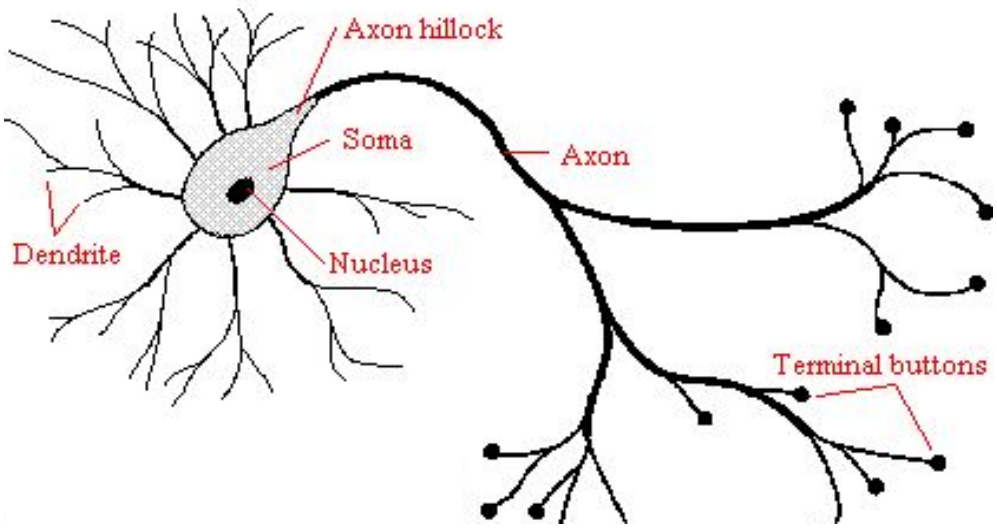


Fig. 3.1. Biological nervous system [37]

When the neuron receives excitatory input, which is sufficiently large compared to inhibitory input, it sends a spike of electric signal down to the axon.

3.4. Artificial Neuron

The basic processing unit of artificial neural network is artificial neurons, which simulates the same basic four functions of the biological nervous system. Artificial neuron is much simpler than biological neuron. The figure of artificial neuron is shown in Fig. 3.2

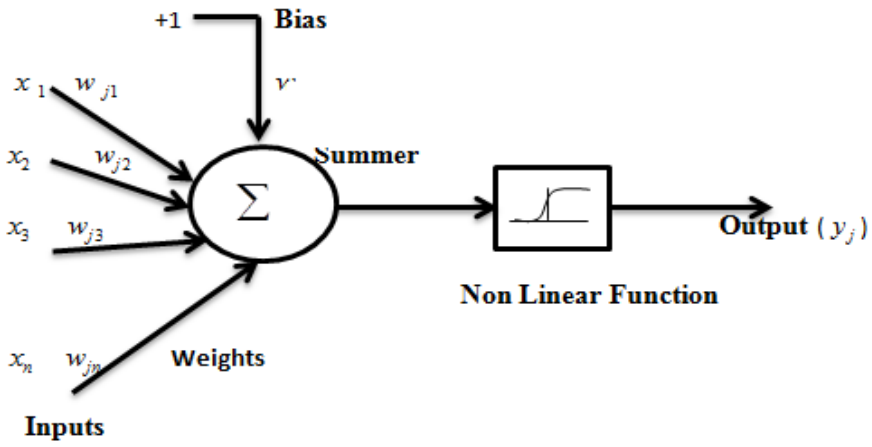


Fig. 3.2. Artificial neuron [15]

Input to the artificial neuron in Fig. 1.16 is represented by input vector i.e., $\vec{X} = \{x_1, x_2, x_3, ...x_n\}$ where each of the input elements are multiplied by corresponding input weights of the neuron. The input weight vector of artificial neuron is given as $\vec{W} = \{w_{j1}, w_{j2}, w_{j3,}.....w_{jn}\}$. In general, the sum of the weighted input signal is fed to an activation function (linear or nonlinear function) of the neural network to obtain the response of the network.

3.5. Types of Activation Functions in Neural Networks

The various activation functions used in neural networks are explained below

3.5.1. Identity Function

The function is given by

$$\phi(x) = x, \ \forall x \tag{3.1}$$

This is shown in Fig. 3.3

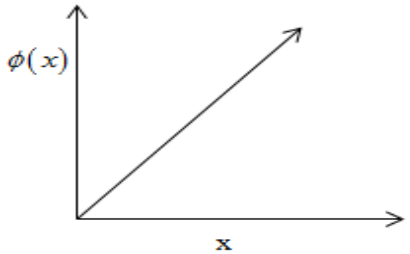


Fig. 3.3. Identity function [37]

3.5.2. Binary Step Function

This function is given by

$$\phi(x) = \begin{cases} 1; & if\ \phi(x) \geq \theta \\ 0; & if\ \phi(x) < \theta \end{cases} \tag{3.2}$$

The binary step function is also known as McCulloch-Pitts neuron model. The figure of binary step function is shown in Fig. 3.4

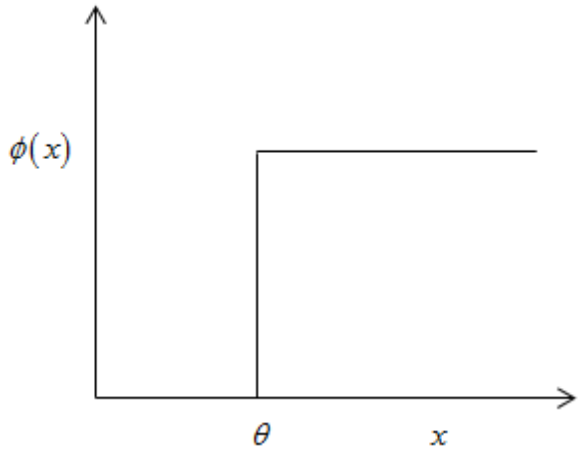


Fig. 3.4. Binary step function [37]

3.5.3. Sigmoidal Function

This is basically an S-shaped curve. These are used in multilayer network like back propagation and radial basis function, etc. These are of two types.

3.5.3.1. Binary Sigmoidal Function

It is also known as logistic function. It ranges between 0 and 1. The mathematical formula of binary sigmoidal function is given below

$$\phi(x) = \frac{1}{1+\exp(-\sigma x)} \tag{3.3}$$

where σ is the slope parameter. The figure of binary sigmoidal function is shown in Fig. 3.5.

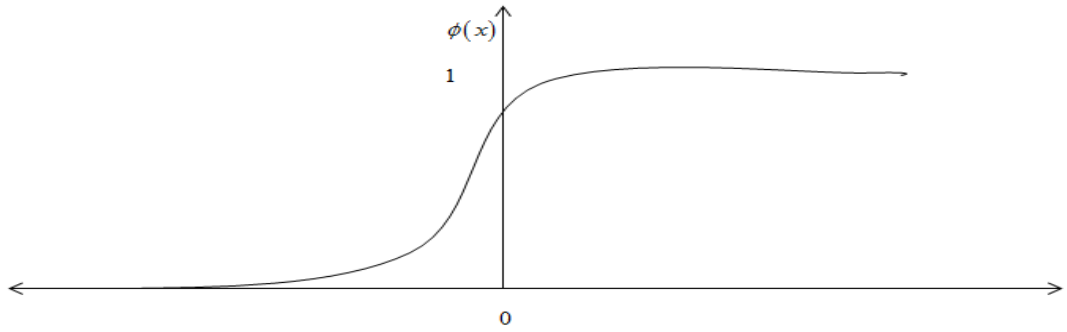


Fig. 3.5. Binary sigmoidal function [15]

3.5.3.2. Bipolar Sigmoidal Function

The desired range of bipolar sigmoidal function is in between +1 and -1. The bipolar sigmoidal function is given as

$$b(x) = 2\phi(x) - 1 \tag{3.4}$$

The figure of bipolar sigmoidal function is shown in Fig. 3.6.

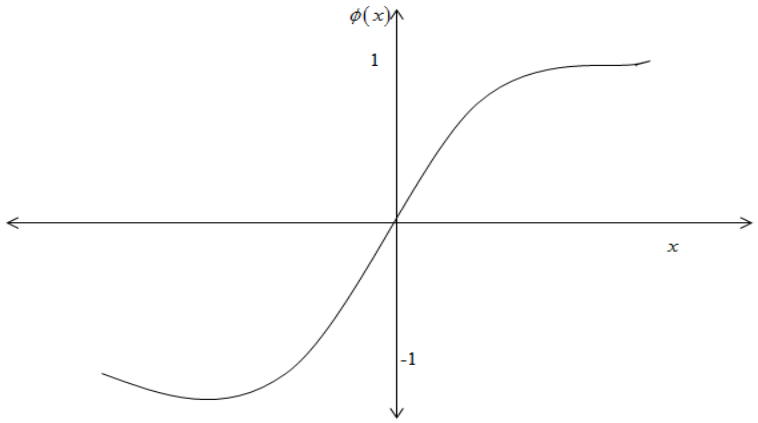


Fig. 3.6. Bipolar sigmoidal function [37]

3.6. Layers

Artificial Neural Network is a simple cluster of artificial neurons. This clustering occurs by making layers, which are interconnected to one another. Generally, ANN has a similar structure of the topology. In the layers, neurons in the input layer receive the input from the real world and neurons in the output layer give the output to the external environment. All the remaining neurons in the network are the hidden neurons in the hidden layer. The architecture of three layers of artificial neural network is shown in Fig. 3.7.

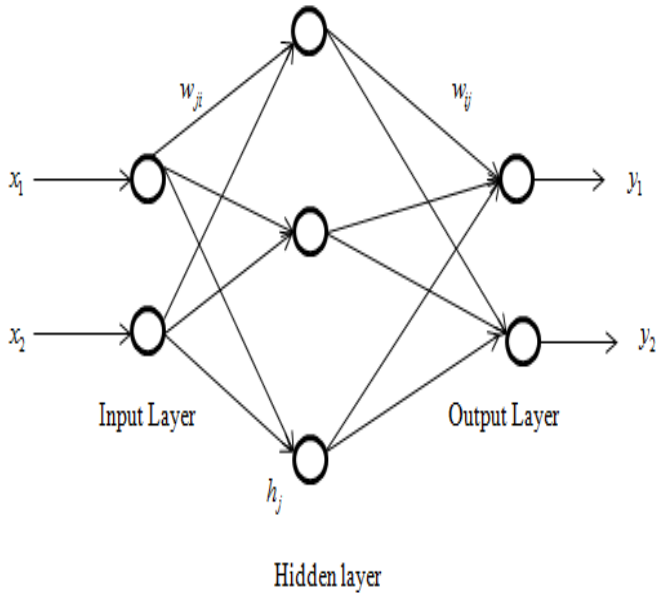


Fig. 3.7. Three layer architecture for ANN [40]

As the figure proves that the neurons are grouped into layers. The input layer consists of neurons that receive the input from the real world and the output layer consists of neurons that communicate the network output to external environment. Generally, the hidden layer is sandwiched between the input layer and the output layer as shown in Fig. 3.7. The input layer receives the input from one source and its neurons produce the output, which further becomes the input to the other layers of the network. This process is continued until the desired condition is satisfied. The number of the hidden neurons is determined through the method of trial and error. A training set of data is used to memorize the network, which further useful for the test data sets.

3.7. Types of Connections

Neurons are connected via a path which carries the output from one neuron as input to another neuron. These paths are basically unidirectional, but may be two-way connection between the neurons.

3.7.1. Feedforward Connection

It allows the signal to travel only in one direction, i.e. from input to output layer. There is no feedback loop; it means the output of any layer does not affect the same layer. It is basically used for the pattern recognition. The figure of the feed forward connection [38] is shown in Fig. 3.8.

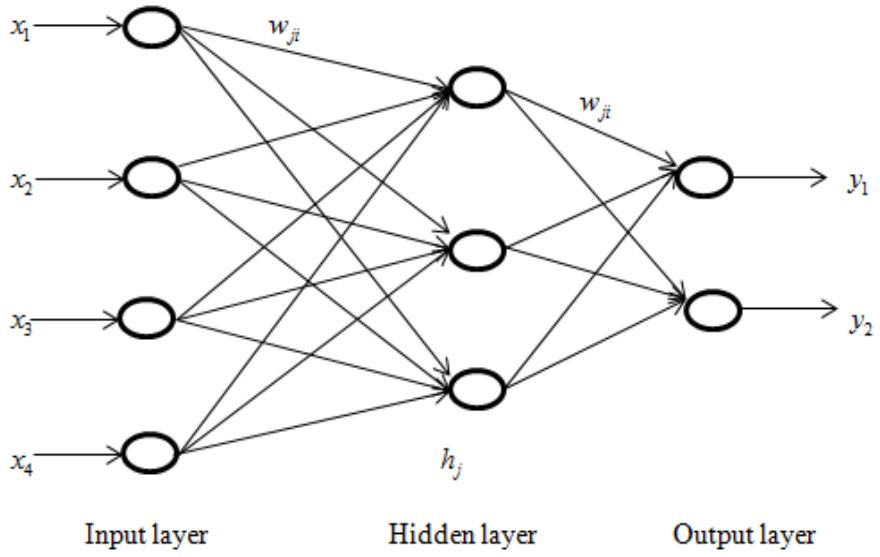


Fig. 3.8. Feedforward connection [40]

3.7.2. Feedback Connection

The input layer neurons receive the input from the real world and send their output to the output layer; also they receive the input back from the neuron of the second layer through the feedback loops in the network. Feedback networks [39] change their state dynamically; until they achieve the desired equilibrium point. They remain in a desired equilibrium point until the input changes and new equilibrium need to be found. The figure of the feedback network in a neural network is shown in Fig. 3.9.

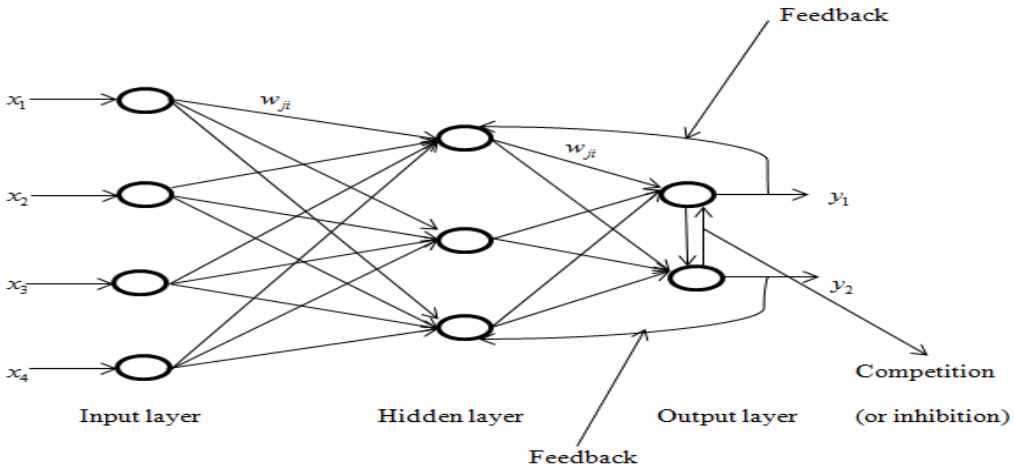


Fig. 3.9. Feedback connection [39]

3.8. Learning

The artificial neural network learns about its environment through an interactive process of the adjustment of its free parameters i.e. synaptic weight and the bias level. The type of learning is depending upon how the free parameter changes take place. The set of rules in an artificial neural network is used to solve a particular learning problem is called as learning algorithm. Each learning algorithm is different from the others algorithm on the way how the adjustment of the free parameters takes place in an artificial neuron. The strength of the connection between the neurons is stored in free parameters of the particular connection. The network learns new knowledge by adjusting the free parameters.

3.9. Training Methods used in Artificial Neural Network

The various training method used in artificial neural network is given below

3.9.1. Supervised Training

In this training, a set of sample input data is provided to the network and compares the network output to the desired response. The training is continued until the network output is able to provide the desired response. In an artificial neural network, a sequence of training vectors exist with target output vectors. The synaptic weights of the neural network are adjusted according to the learning algorithm. This process of training is known as supervised training. In a logic circuit, the target output as '+1' if the logic condition is satisfied or '0' if the logic condition is not satisfied. This type of logical network is trained with the supervised algorithm. Some of the supervised learning algorithms include Hebbian network, Pattern association memory, network and Back propagation network etc.

3.9.2. Unsupervised Training

In this training process, the training input vectors have not known the target output. The network adjusts the free parameters so that the most similar input vectors are assigned to the same output unit. Unsupervised networks are more complex and difficult to implement it. It involves the feedback connections or the iterative process until the suitable result is not obtained. Unsupervised networks are also known as self-organizing networks. This training method is adopted into the self-organizing maps (SOM) and adaptive resonance theory etc.

3.10. Learning Rules

There are a number of learning rules in an artificial neural network which are used to update synaptic weights and bias level between the artificial neurons. Some of the learning rules in an artificial neural network are explained below

3.10.1. Hebbian Learning Rule

The Hebbian learning rule is the most famous from all the learning rules. It states that, "when an axon of cell A is to excite the cell B and takes part to fire it, then some metabolic growth changes takes place such that the A's efficiency to fire B has increased and vice-versa". This learning is also known as correlation learning.

This statement is divided into two parts:

1. If the two neurons i.e., pre-synaptic neuron and post-synaptic neuron are activated simultaneously, then the strength of the synaptic weights has increased.

2. If the two neurons i.e., pre-synaptic neuron and post-synaptic neuron is not activated simultaneously, then the strength of the synaptic weights are weakened or eliminated.

The simplest form of the Hebbian learning rule to update the synaptic weights in an artificial neural network is given below

$$\Delta w = x_i y \tag{3.5}$$

where x_i is input vector and y is response of the network.

The Hebbian learning rule is purely for feed forward and an unsupervised learning. It states that if the cross product of the input and output is positive, and then there is strengthening in the synaptic weights otherwise weakened.

3.10.2. Delta Learning Rule

The Delta learning rule is also called as a Widrow-Hoff rule [37]. It is only applicable for the supervised training mode. It stated that, "The adjustment of the synaptic weights is proportional to the product of the error signal and the input signal of the synapses". The objective of the delta rule is to minimize the error for all the training patterns and maximize the performance of a network.

The Delta learning rule is applicable for single output neuron or more than one single output neuron. The derivations of a single output neuron and several output neuron is given below

3.10.2.1. Delta Rule for Single Output Neuron

Delta learning rule [37] changes the synaptic weights in order to minimize the difference between the network response, y_{in} and the target response t.

The delta learning rule is given by

$$\Delta w_i = \eta\left(t_j - y_{-inj}\right)x_i \tag{3.6}$$

where x_i is the input vector, y_{-inj} is the network input to the activation function, t_j is the target vector and η is the learning rate.

The Mean Square Error (MSE) of a particular training pattern is

$$E = \sum_J \left(t_j - y_{-inj}\right)^2 \tag{3.7}$$

The gradient of E is found by the partial derivative of the E with respect to their synaptic weights. The error is minimized by adjusting the synaptic weights w_{IJ}

Take the partial differential of E with respect to w_{IJ}

$$\frac{\partial E}{\partial w_{IJ}} = \frac{\partial}{\partial w_{IJ}} \sum_i (t_j - y_{inj})^2 \tag{3.8}$$

The synaptic weights w_{IJ} influence the error only at output neuron y_J .

Also, we know that

$$y_{inJ} = \sum_{i=1}^{n} \left(t_j - y_{inj}\right)^2 \tag{3.9}$$

Hence, we get

$$\frac{\partial E}{\partial w_{IJ}} = \frac{\partial\left(t_j - y_{inj}\right)^2}{\partial w_{IJ}} \tag{3.10}$$

$$\frac{\partial E}{\partial w_{IJ}} = 2\left(t_j - y_{inj}\right)(-1)\frac{\partial y_{inj}}{\partial w_{IJ}} \tag{3.11}$$

$$\frac{\partial E}{\partial w_{IJ}} = -2\left(t_j - y_{inJ}\right)x_I \tag{3.12}$$

Therefore the error is rapidly reduced by adjusting the synaptic weights between the artificial neurons from the I^{th} input neuron to the J^{th} output neuron according to the delta rule [37] which is given by

$$\Delta w_{IJ} = \eta\left(t_J - y_{inJ}\right)x_I \tag{3.13}$$

3.10.2.2. Delta Rule for Several Output Neurons

The Mean Squared Error (MSE) of particular training pattern [37] is given as

$$E = \sum_J \left(t_j - y_j\right)^2 \tag{3.14}$$

where E is the function of all the synaptic weights.

The gradient of E is found by the partial derivative of E with respect to their each synaptic weight. The error is reduced by adjusting the synaptic weights w_{IJ} in the direction of $\frac{-\partial E}{\partial w_{IJ}}$

Differentiate E partially with respect to the synaptic weights w_{IJ}

$$\frac{\partial E}{\partial w_{IJ}} = \frac{\partial}{\partial w_{IJ}} \sum_J \left(t_j - y_j\right)^2 \tag{3.15}$$

The synaptic weights w_{IJ} influence the error only at output neuron y_J

Also, we know that

$$y_{inJ} = \sum_{i=1} x_i w_{iJ} \tag{3.16}$$

$$y_J = \phi(y_{inj}) \tag{3.17}$$

From eq. (3.15), we get

$$\frac{\partial E}{\partial w_{IJ}} = 2\left(t_j - y_j\right)(-1)\frac{\partial y_j}{\partial w_{IJ}} \tag{3.18}$$

$$\frac{\partial E}{\partial w_{IJ}} = 2\left(t_j - y_{inj}\right)\frac{\partial \phi\left(y_{inJ}\right)}{\partial w_{IJ}} \tag{3.19}$$

$$\frac{\partial E}{\partial w_{IJ}} = 2\left(t_j - y_j\right)\left(x_i\right)\phi'\left(y_{inJ}\right) \tag{3.20}$$

Therefore the error is rapidly reduced by adjusting the synaptic weights between the artificial neurons from the I^{th} input neuron to the J^{th} output neuron according to the delta rule [37] which is given by

$$\Delta w_{IJ} = \eta\left(t_j - y_j\right)\left(x_i\right)\phi'\left(y_{inJ}\right) \tag{3.21}$$

3.11. Application of Neural Networks

Artificial Neural Network has a number of applications such as interpretation, prediction, planning and one of the most important and successful application of neural network is the clustering of data and pattern recognition.

Another application of the artificial neural network is character recognition and handwritten recognition. This area uses credit card processing and banking where reading and correct recognition of documents is crucial significance.

Other important applications of neural networks are biometrics, communication, digital image processing, speech processing, business, robotics, prediction of the stock market, heart disease database using vector quantization artificial neural network optimization, data filtering, and data association and to intrusion detection etc.

3.12. Image Compression using Neural Network

The neural network is one of the most important techniques which is used for the image compression apart from the Motion Picture Expert Group (MPEG) and Joint Photo Expert Group (JPEG) because of the noise suppression, fault tolerance, high compression and adaptive learning capabilities. Some of the neural network techniques used for image compression are explained below

1. Backpropagation Neural Network.
2. Hebbian Learning based Neural Network.

3.12.1. Backpropagation Neural Network

The figure of the basic Back-propagation [40] neural network is shown in Fig. 3.10. In this structure, there are three layers, one input layer, one hidden layer and one output layer. The input and the output layers are fully interconnected to the hidden layer. Compression of the image is achieved by assigning the less number of hidden neurons (K) in the hidden layer as compared to the input and the output layers. The input image is divided into non-overlapping blocks of size 4×4, 8×8 or 16×16 . When the input vector is considered N-dimensional which is equal to the number of pixels in each block,

then the synaptic weight connected to each neurons in the hidden layer is represented by {w_{ji}, j=1, 2, 3, …….K and i=1, 2, 3, ……N} which is also written as in the matrix form $K \times N$. The connection from the hidden layer to the output layer is another weight matrix which is written as $N \times K$. Compression of the image is achieved by training the neural network in such a way that the synaptic weights $\{w_{ji}\}$ scaled the input vector of N-dimension to the K-dimension (K<N) and produce the minimum square error between the input and the output. The linear operation of the Fig. 3.10. is shown as

For Encoding

$$h_j = \sum_{i=1}^{N} w_{ji} x_i \tag{3.22}$$

For Decoding

$$\hat{x} = \sum_{j=1}^{K} w_{ij} h_j \tag{3.23}$$

Where $x_i \in [0,1]$ denotes the normalized pixel value for grayscale images whose gray scale levels are [0,255]. The pixel is normalized because the neural network work efficiently when both input and output are limited to range [0,1]. The above linear function is also converted into non-linear network when the nonlinear activation function is used in the hidden and the output layer. Basic Back-propagation network compressed the image into two phases, i.e. training and encoding. In the first phase, a set of the image pixels is designed to train the network through the back-propagation learning rule, which uses the input pixels as the desired response. This is similar to the compression of input data into the narrow channel by the hidden layer neurons and reconstructs the input from the hidden layer neurons to the output layer neurons.

3.12.2. Hebbian Learning based Neural Network

The Hebbian learning rule [40] comes from the Hebb's postulates that if pre-synaptic neuron and post-synaptic neuron is active simultaneously, then the strength between the neuron has increased and vice versa. Hence the output of the neuron is given as $[h] = [w]^T [x]$ and the learning rule is described as

$$w_i(n+1) = \frac{w(n) + \eta h_i(n) x(n)}{\| w_i(n) + \eta h_i(n) x(n) \|} \tag{3.24}$$

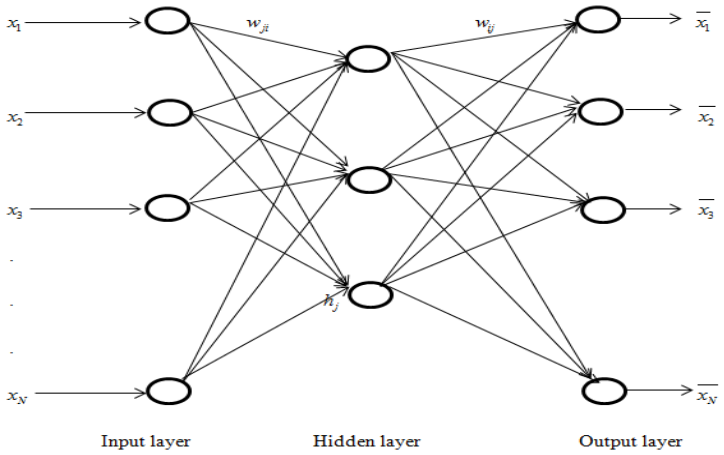


Fig. 3.10. Backpropagation neural network [40]

Hence the output of the neuron is given as $[h]=[w]^T[x]$ and the learning rule is described as

$$w_i(n+1)=\frac{w(n)+\eta h_i(n)x(n)}{\| w_i(n)+\eta h_i(n)x(n)\|} \tag{3.24}$$

where $w(n+1)$ is the updated weights for each iteration (n+1), $w(n)$ is the old weights, η is the learning rate, $h_i(n)$ is the i^{th} output value and $\|.\|$ denote the Euclidean norm which is used to normalize the updated weights and make the learning stable. Oja is the scientist who proposed linearized Hebbian learning rule and expanding the series of updating synaptic weights between the neuron which is given below

$$w_i(n+1)=w_i(n)+\eta\left[h_i(n)x(n)-h_i^2(t)w_i(n)\right] \tag{3.25}$$

To obtain the particular l principal components, Sanger is the scientists who extend the above model which removes the effects of previous principal components through the Gram-Schmidt orthogonalization and make the synaptic weights converge to the desired Eigen vectors. And hence the image is compressed.

Chapter 4

METHODLOGY AND ANALYSIS

This chapter discuss Principal Component Analysis techniques named as Singular Value Decomposition (SVD), Gram Schmidt Procedure (GSP) and Generalized Hebbian Algorithm (GHA) along with their mathematical analysis.

4.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [15] is a statistical method to transform a correlated data set into a smaller number of data set called as principal components. The principal component analysis uses the vector space transform technique to reduce the dimensionality of the correlated data sets. By using the statistical projection, the original data set which included a large number of variables is represented with few numbers of variables called as principal components. Hence, it is easy to examine the outliers and patterns in the transform dataset which is not feasible without performing the PCA.

4.2. Mathematical Analysis of Principal Component Analysis

PCA takes a correlated or large data set and identified a new optimum basis to re-express the data. Consider an input image X which is represented in the form of matrix of size $m \times n$, where n is the columns and r is the rows of the input image matrix X. We have to transform this input matrix X, into another matrix X_{new} of dimension $m \times n$, so that some matrix T of dimension $m \times m$ i.e.,

$$X_{new} = TX \tag{4.1}$$

Eq. (4.1) represents the change of basis. The rows of T matrix is assumed in the form of row vectors, i.e. $t_1, t_{2,} t_3, ..., t_m$ and the columns of the input data matrix are represented in the column vectors i.e. $x_1, x_2, x_3, ..., x_n$ then the eq. (3.1) is re-written as

$$TX = \begin{pmatrix} t_1.x_1 & \cdots & t_1.x_n \\ \vdots & \ddots & \vdots \\ t_m.x_1 & \cdots & t_m.x_n \end{pmatrix} = X_{new} \tag{4.2}$$

where $t_{i,} x_j \in \mathbb{R}^m$ and $t_i . x_j$ is the inner product. This inner product tells that the original input data matrix X is projected onto the columns of T matrix. Hence, the rows of T, $t_1, t_2, t_3, ..., t_m$ are the new basis for representing the columns of X. And the rows of T become our principal component directions. Now our main focus is what should be the

new basis, how the input image X on the new basis or in other words, how to define the independence between the principal component directions on the new basis. PCA defines the independence by considering the variance of data in the original basis. It de-correlates the original data by finding the directions in which the variance have been maximized and uses these directions are the new basis. The variance of the original data is given below

$$\sigma_A^2 = E\left[\left(A-\mu\right)^2\right] \tag{4.3}$$

where A is the random variable and μ is the mean of the original image. Suppose we have a vector of n discrete measurements, $\tilde{c} = \left(\tilde{c}_1, \tilde{c}_{2,}\tilde{c}_3, ...\tilde{c}_n\right)$ with mean μ_c. If we subtract the mean of each measurement then we obtain the zero mean measurements i.e., $c = \left(c_1, c_2, c_3, ..., c_n\right)$. Hence the variance of these measurements is given below

$$\sigma_c^2 = \frac{1}{n} cc^T \tag{4.4}$$

Second vector of n measurement with zero mean is represented by $w = \left(w_1, w_{2,} w_3, ..., w_n\right)$. We can generalize this idea and obtain the covariance of c and w. Covariance is defined how the two variables are changed together. The covariance of the first vector (c) and second vector (w) is given below

$$\sigma_{cw}^2 = \frac{1}{n-1} cw^T \tag{4.5}$$

The original input image X is represented in m rows and each of the length and i.e.

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \in \mathbb{R}^{mxn}, x_i^T \in \mathbb{R}^n \tag{4.6}$$

Since in eq. (4.6) we have a row vector of each variable, each of these vectors contains the sample of one particular variable. For example, x_i is a vector of n samples for i^{th} variable. Hence it is consider the following matrix product

$$C_X = \frac{1}{n-1} XX^T = \frac{1}{n-1} \begin{pmatrix} x_1 x_1^T & \cdots & x_1 x_m^T \\ \vdots & \ddots & \vdots \\ x_m x_1^T & \cdots & x_m x_m^T \end{pmatrix} \in \mathbb{R}^{mxm} \tag{4.7}$$

Eq. (4.7) is known as covariance matrix in which the diagonal elements are variances and the non-diagonal elements are the covariance. Covariance tells that how two variables are correlated. In the Principal Component Analysis (PCA) method the variables in the

transform domain are de-correlated as possible. This is similar to saying that the covariance of different variables in the matrix $C_{X_{new}}$ is close to zero as possible. To construct the $C_{X_{new}}$ matrix, the following requirements are needed

1 Maximize the variances.

2 Minimize the non-diagonal entries.

Thus, our main objective is to choose the transformation matrix T in such a way that the $C_{X_{new}}$ matrix should be diagonal.

We have to make an assumption that vector in new basis $t_1, t_2, t_3, ...t_l$ are orthogonal to each other. Thus the covariance matrix in transform domain is given by

$$C_{X_{new}} = \frac{1}{n-1} X_{new} X_{new}^T \tag{4.8}$$

$$C_{X_{new}} = \frac{1}{n-1} (TX)(TX)^T \tag{4.9}$$

$$C_{X_{new}} = \frac{1}{n-1} TRT^T \tag{4.10}$$

where $R = XX^T$ is a symmetric matrix. And we know from the linear algebra that every symmetric matrix is orthogonal and diagonalizable. Hence the R can be re-written as

$$R = QDQ^T \tag{4.11}$$

where Q is an $m \times n$ orthonormal matrix whose columns are the principal components of the R matrix and D is the diagonal matrix which has the variance value of R across the diagonal elements. By choosing the rows of T are the principal components of R, such that T= Q^T and vice versa. Now eq. (4.10) can be re-written as

$$C_{X_{new}} = \frac{1}{n-1} Q^T \left(QDQ^T\right) Q \tag{4.12}$$

$$C_{X_{new}} = \frac{1}{n-1} Q^T \left(QDQ^T\right) Q \tag{4.13}$$

Since Q is an orthonormal matrix so $Q^T Q = I$ where I is the identity matrix of dimension m x m.

Hence the choice of T is given as

$$C_{X_{new}} = \frac{1}{n-1} D \tag{4.14}$$

The largest eigenvalue corresponds to the first eigenvector, second largest to the second largest eigen vector and so on. Hence we obtain our objective to diagonalise the covariance matrix of the transform data. The problems associated with the classical method to solve the eigen values and finding the corresponding eigen vectors to it and then project the input image onto the eigen vector space is solved by using a single linear neuron model which is based on the Hebbian learning mechanism [11]. It states, "If both the pre-synaptic signal and the post-synaptic signals are correlated to each other than the synaptic weights are strengthened and vice versa". The single linear neuron model is shown in Fig. 4.1. Which extracts the first principal component of the correlation matrix R. The output of the single neuron model y, is the weighted sum of the synaptic weight $\{w_1, w_2, w_3, ..., w_m\}$ and the inputs $\{x_1, x_2, x_3, ..., x_m\}$ or in the form of vectors is given below

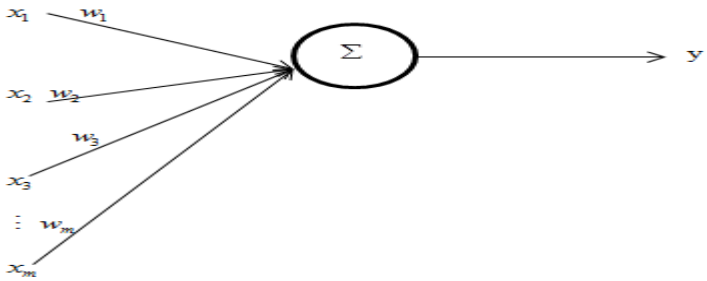


Fig. 4.1. Single linear neuron model [15]

A simple Hebbian learning rule to update the weights is nothing but it is the product of the input and the output values as

$$w(n+1) = w(n) + \eta y(n) x(n) \tag{4.15}$$

where η is the learning rate and n is the iteration number. However, such learning rule is unstable because the weights are increased without any boundaries. To stable the learning rule, we normalized the synaptic weights at each step which is given below

$$w(n+1) = \frac{w(n) + \eta y(n) x(n)}{\| w(n) + \eta y(n) x(n) \|} \tag{4.16}$$

where $\|.\|$ is the Euclidean norm. Oja linearized the above rule using series and form as

$$w(n+1) = w(n) + \eta\left[y(n)x(n) - y^2(n)w(n) \right] \tag{4.17}$$

This eq. (4.18) converges to the first principal component of the correlation matrix R.

4.3. Singular Value Decomposition (SVD)

The input image matrix X of size $n \times m$ is given as $A \in \mathbb{R}^{n \times m}$, which is represented in the form of Singular Value Decomposition [41] as

$$A = U \sum V^T \tag{4.18}$$

Where $U \in \mathbb{R}^{n \times n}$ is orthonormal matrix, $\sum \in \mathbb{R}^{n \times m}$ is diagonal matrix and $V \in \mathbb{R}^{m \times m}$ is orthonormal matrix.

In the diagonal matrix, the diagonal entries are represented by σ_i which are non-negative and hence called the singular values of image matrix A. They are sorted in such a way that the largest singular value σ_1 is placed at the point (1,1) of $\sum$ and other singular values are placed down the diagonal of $\sum$ and satisfy the $\sigma_1 \geq \sigma_2 \geq .....\sigma_p \geq 0$. Since $V \in \mathbb{R}^{m \times m}$ and $U \in \mathbb{R}^{n \times n}$ are the orthonormal matrix, hence their column form basis for the vector $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively. Therefore, a vector $x \in \mathbb{R}^m$ can be expanded to the basis formed by the columns of V and any vector $b \in \mathbb{R}^n$ can be expanded to the basis formed by the columns of U. The vectors for these expansions are $\hat{x}$ and $\hat{b}$ which is represented as

$$\hat{b} = U^T b \tag{4.19}$$

and

$$\hat{x} = V^T x \tag{4.20}$$

If equation $b = Ax$ satisfies, then the eq. (4.20) is re-written as

$$U^T b = U^T A x \tag{4.21}$$

$$\hat{b} = U^T \left(U \sum V^T \right) x \tag{4.22}$$

$$\hat{b} = \sum \hat{x} \tag{4.23}$$

Hence, SVD allows every matrix to be diagonal. Therefore, we choose the appropriate normal basis. We know that the every non-singular values of A are the square roots of the non-zero eigen values of the $A^T A$ or AA^T. To obtain the original input image matrix, we define a new matrix Z of a size $n \times m$ which is given below

$$Z = \frac{1}{\sqrt{n-1}} X^T \tag{4.24}$$

We subtract the row mean from each entry in eq. (4.21) to obtain the zero mean across the rows. Hence the new matrix Z has the columns with zero mean. Consider the following $m \times m$ matrix, $Z^T Z$

$$Z^T Z = \frac{1}{n-1} X X^T \tag{4.25}$$

i.e.,

$$Z^T Z = C_X \tag{4.26}$$

The eq. (4.26) represents the covariance matrix of X, C_X . If we perform the SVD of the matrix $Z^T Z$, the principal components are the columns of the orthogonal matrix, V. The original input image matrix X, is projected onto the direction of the principal components V by following equation

$$Y = V^T X \tag{4.27}$$

The original input image matrix X is recovered by following equation

$$X = VY \tag{4.28}$$

4.4. Gram Schmidt Orthogonalization Procedure (GSP)

The Gram Schmidt Procedure [42] is used to obtain the orthonormal basis from the non-orthonormal basis. For example, we have a linearly independent image data in row vector (V) which are not orthonormal i.e. set of functions . To make an orthonormal set $\{\phi_1, \phi_2, ...\}$ from this set, we proceed as follows. First, obtain ϕ_1 by simply normalize x_1 which is given below

$$\phi_1 = \frac{x_1}{\| x_1 \|} \tag{4.29}$$

Next, obtain ϕ_2 which must be ensured that it is orthogonal to ϕ_1 and then normalize it. Take a function ψ_2 which is represented in the form as

$$\psi_2 = x_2 + c_{12}\phi_1 \tag{4.30}$$

such that the inner product ϕ_1 and ψ_2 i.e., $\langle \phi_1 | \psi_2 \rangle$ is equal to zero. Then we can set $\phi_2 = \frac{\psi_2}{||\psi_2||}$. Taking the dot product of eq. (4.30) on both sides with ϕ_1, we obtain as

$$0 = \langle \phi_1 | x_2 \rangle + c_{12} \langle \phi_1 | \phi_1 \rangle \tag{4.31}$$

Because ϕ_j are orthonormal, it says that

$$c_{12} = -\langle \phi_1 | x_2 \rangle \tag{4.32}$$

Therefore, eq. (4.30) is written as

$$\psi_2 = x_2 - \langle \phi_1 | x_2 \rangle \phi_1 \tag{4.33}$$

$$\phi_2 = \frac{\psi_2}{||\psi_2||} \tag{4.34}$$

Now we obtain two sets of functions which are orthonormal to each other. Next, obtain ϕ_3 which must be ensured that it is orthogonal to both ϕ_1 and ϕ_2, then normalize it. Take a function ψ_3 which is represented as

$$\psi_3 = x_3 + c_{13}\phi_1 + c_{23}\phi_2 \tag{4.35}$$

such that $\langle \phi_1 | \psi_3 \rangle = \langle \phi_2 | \psi_3 \rangle = 0$. Then set $\phi_3 = \frac{\psi_3}{||\psi_3||}$. Take the dot product of ϕ_1 and ϕ_2 separately on both sides of eq. (4.35), we get

$$0 = \langle \phi_1 | x_3 \rangle + c_{13} \langle \phi_1 | \phi_1 \rangle + c_{23} \langle \phi_1 | \phi_2 \rangle \tag{4.36}$$

$$0 = \langle \phi_2 | x_3 \rangle + c_{13} \langle \phi_2 | \phi_1 \rangle + c_{23} \langle \phi_2 | \phi_2 \rangle \tag{4.37}$$

Because ϕ_j are orthonormal; it follows that

$$c_{13} = -\langle \phi_1 | x_3 \rangle, \qquad c_{23} = -\langle \phi_2 | x_3 \rangle$$

Therefore eq. (4.35) can be written as

$$\psi_3 = x_3 - \langle \phi_1 | x_3 \rangle \phi_1 - \langle \phi_2 | x_3 \rangle \phi_2 \tag{4.38}$$

$$\phi_3 = \frac{\psi_3}{||\psi_3||} \tag{4.39}$$

Hence, we get three set of functions which are orthonormal.

Continuing this process, we can get each function ϕ_j as follow

$$\psi_j = x_j - \sum_{k=0}^{j-1} \langle \phi_k | x_j \rangle \phi_k \tag{4.40}$$

$$\phi_j = \frac{\psi_j}{\| \psi_j \|} \tag{4.41}$$

This set of functions $\{\phi_1, \phi_2, ...\}$ is an orthonormal basis of vector $\{x_1, x_2, ...\}$ with respect to the dot product is used.

4.5. Generalized Hebbian Algorithm (GHA)

Sanger GHA [11] extends the Oja learning rule model to determine the first l principal components from the correlation matrix R which is similar to the first data by removing the previous principal components through the Gram Schmidt procedure. Fig. 4.2 shows the structure of such system. Each output y_i of the defined structure is the i^{th} principal component unit. In vector, it is written as

$$y = Wx \tag{4.42}$$

with $y \in \mathbb{R}^l, W \in \mathbb{R}^{l \times m}$ and $l \leq m$.

In other words, Sanger Generalized Hebbian Algorithm updates the synaptic weights by following learning rule

$$w_{ji}(n+1) = w_{ji}(n) + \eta y_j(n) \left[x_i(n) - \sum_{k=1}^{j} w_{ki}(n) y_k(n) \right], \text{ for } i = 1, 2, 3, ..., m \text{ and } j = 1, 2, 3, ..., l \tag{4.43}$$

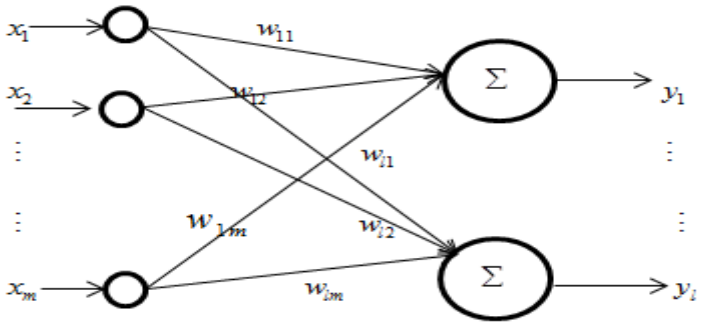


Fig. 4.2. Principal components, linear neural network model [15]

where j is the output neuron, n is the iteration number, x_i is the input vector and η is the learning rate.

Eq. (4.43) can be re-written as

$$w_{ji}(n+1) = w_{ji}(n) + \eta y_j(n)\left[x_i^{'}(n) - w_{ji}(n)y_j(n)\right],\ for\ i = 1,2,3,.....,m\ and\ j = 1,2,3,.....,l \tag{4.44}$$

$$x_i^{'}(n) = x_i(n) - \sum_{k=1}^{j-1} w_{ki}(n)y_k(n) \tag{4.45}$$

Eq. (4.44) can be re-written as

$$w_{ji}(n+1) = w_{ji}(n) + \eta y_j(n)x_i^{''}(n) \tag{4.46}$$

where $x_i^{''}(n) = x_i^{'}(n) - w_{ji}(n)y_j(n)$ (4.47)

Eq. (4.45) can be represented in signal flow as

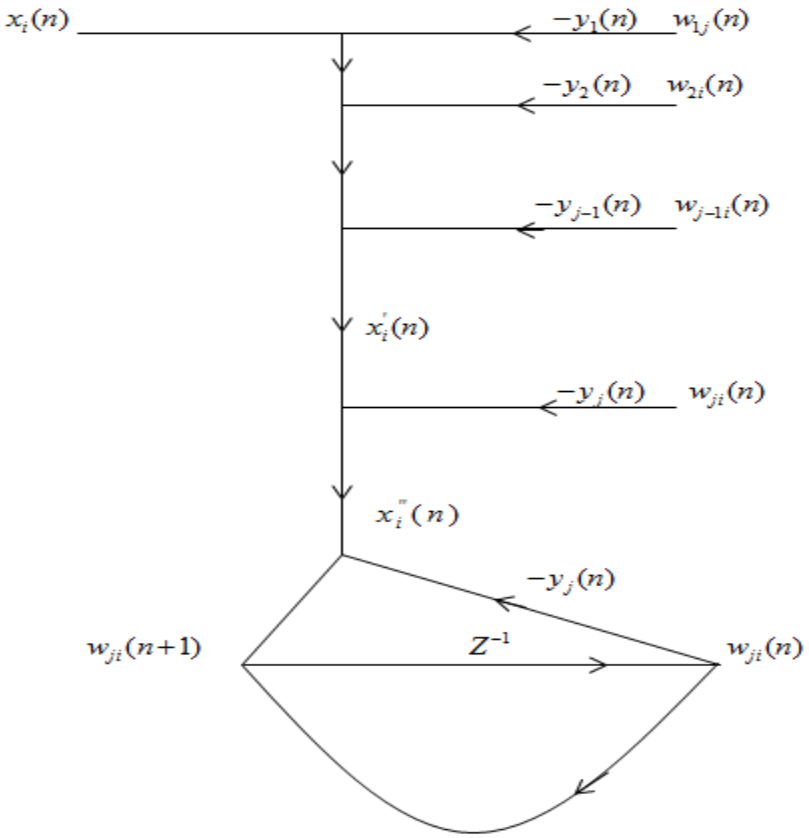


Fig. 4.3. Signal flow of GHA

For $i = 1,2,3,.....,m$ then $\triangle w_{j1,} \triangle w_{j2}, \triangle w_{j3}, ..., \triangle w_{jm} = \triangle \vec{w}_j(n)$

Eq. (4.44) can be written as in the matrix form as

$$w_j(n+1) = w_j(n) + \eta y_j(n)\vec{x}_i^{'}(n) - \eta y_j^2(n)\vec{w}_j(n), \quad \text{for } j = 1,2,3,.....,l \tag{4.46}$$

where $\vec{x}_i^{'}(n) = \vec{x}_i(n) - \sum_{k=1}^{j-1} \vec{w}_k(n) y_k(n)$

The following observations are made from eq. (4.44)

1. For the first neuron, j=1

$$\vec{x}^{'}(n) = \vec{x}(n) - \vec{w}_1(n) y_1(n) \tag{4.47}$$

$$\Delta\vec{w}_1(n) = \eta y_1(n)\vec{x}_i^{'}(n) - \eta y_1^2(n)\vec{w}_1(n) \tag{4.48}$$

Initially $\vec{w}_1(n) = \vec{w}(n)$ and $y_1(n) = y(n)$

Eq. (4.44) can be re-written as

$$w(n+1) = w(n) + \eta y(n)\left[\vec{x}_i^{'}(n) - y(n)w(n)\right] \tag{4.49}$$

This neuron will extract the first principal component.

2. For the second neuron, j=2

$$\vec{x}_i^{'}(n) = \vec{x}_i(n) - \vec{w}_1(n) y_1(n) \tag{4.50}$$

Where $\vec{x}^{'}(n)$ is the input vector from which the first principal component from the correlation matrix has been removed.

$$w_2(n+1) = w_2(n) + \eta y_2(n)\left[\vec{x}(n) - \vec{w}_1(n) y_1(n) - \vec{w}(n) y(n)\right] \tag{4.51}$$

So, this neuron will extract the first principal component of $\vec{x}^{'}(n)$ which is similar to the second principal component of the original correlation matrix $\vec{x}(n)$. Similarly, we can extract the first l principal component of the original matrix. Hence the Sanger GHA is used to extract the particular number of principal components from the original correlational matrix $\vec{x}(n)$.

4.5.1. Algorithm for Image Compression based on GHA

Following steps are used to compress the digital images using Generalized Hebbian Algorithm (GHA) based on neural networks.

Step 1: Import an image.

Step 2: Divide an image into non-overlapping blocks of the size such as 8x8, 16x16 and 32x32.

Step 3: Calculate the mean value of all the non-overlapping blocks across the whole image.

Step 4: Subtract the Calculated mean value in step 3 from the every pixel of the respective block of the image. And this will get the result of the zero mean block. This process is repeated for the whole image.

Step 5: Determine the covariance matrix of zero mean block. This process is repeated for the whole image.

Step 6: Extract the Principal Components from all non-overlapping covariance blocks using the following equation

$$w_{ji}(n+1) = w_{ji}(n) + \eta y_j(n)\left[x_i(n) - \sum_{k=1}^{j} w_{ki}(n) y_k(n) \right] \tag{4.52}$$

where η is the learning rate, x_i is the input vector and w_{ji} is the synaptic weights from i^{th} neuron to j^{th} neuron.

Step 7: Every original non-overlapping block is projected to the respective extracted number of principal components. The projection of the original block is given by

$$Z = y^{'} * x \tag{4.53}$$

where y is the extracted number of principal components and x is the original input block. The quality of the reconstructed image is better as the number of the principal components is increased.

Step 8: Before reconstructing the compressed image, every non-overlapping blocks across the whole image is converted back into the original basis which is defined below

$$z = y * Z \tag{4.54}$$

Step 9: Add the mean value of the block to the corresponding original basis block. This process is repeated for the whole image. Hence, the decompressed image is obtained at the receiver which is denoted by $\hat{x}$.

Step 10: After obtaining the decompressed image at the receiver, compute the image quality parameters to compare the visual quality of the original image and the reconstructed image. In digital image processing, there are various number of image quality parameters such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Structure Similarity Index (SSIM) by following equations

Mean Squared Error (MSE)

$$MSE(x,\hat{x}) = \frac{1}{L^2}\sum_{i,j=0}^{L-1}\left[x(i,j) - \hat{x}(i,j)\right]^2 \tag{4.54}$$

where L is the intensity level of gray scale image, $x(i,j)$ is the original image and $\hat{x}(i,j)$ is the reconstructed image.

Peak Signal to Noise Ratio (PSNR)

$$PSNR(x,\hat{x}) = 10\log_{10}\left\{\frac{255^2}{MSE(x,\hat{x})}\right\} \tag{4.55}$$

Structure Similarity Index (SSIM)

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{4.56}$$

where μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y and σ_{xy} is the covariance of x and y.

Chapter 5

RESULTS

5.1. Traditional PCA Methods

In this section, we will represent simulation data for image compression using two traditional PCA methods – SVD and GSP. Simulation results for both PCA methods are given in the form of bar graphs and data tables. Reconstructed images are also provided to justify the simulated data.

Table 5.1. Simulated data of reconstructed (Lena) image with variable block size and principal components

Image	Block size	m=	SVD			GSP		
			MSE	PSNR	SSIM	MSE	PSNR	SSIM
		1	163.65	25.99	69.59	158.11	26.14	69.39
Lena	8×8	4	61.27	30.25	85.53	73.04	29.49	83.18
		7	14.45	36.53	96.60	44.06	31.68	89.14
		10	203.32	25.11	69.43	232.11	24.47	67.03
Lena	32×32	15	142.33	26.59	76.56	164.94	25.95	74.61
		17	120.60	27.20	79.37	141.39	26.62	77.60

Table 5.1, represents the simulated data for Lena image. It represents the data in terms of MSE, PSNR and SSIM with respect of variable block size (8×8, 32×32) and principal components (1, 4, 7; 10, 15, 17) respectively. It is analyzed from the table, for the 8×8 and 32×32 that SVD method has an average gain of 4.41% in SSIM, 6.25% in PSNR and -13.02% in MSE over GSP and 2.79% in SSIM, 2.41% in PSNR and -13.40% in MSE over GSP respective block size.

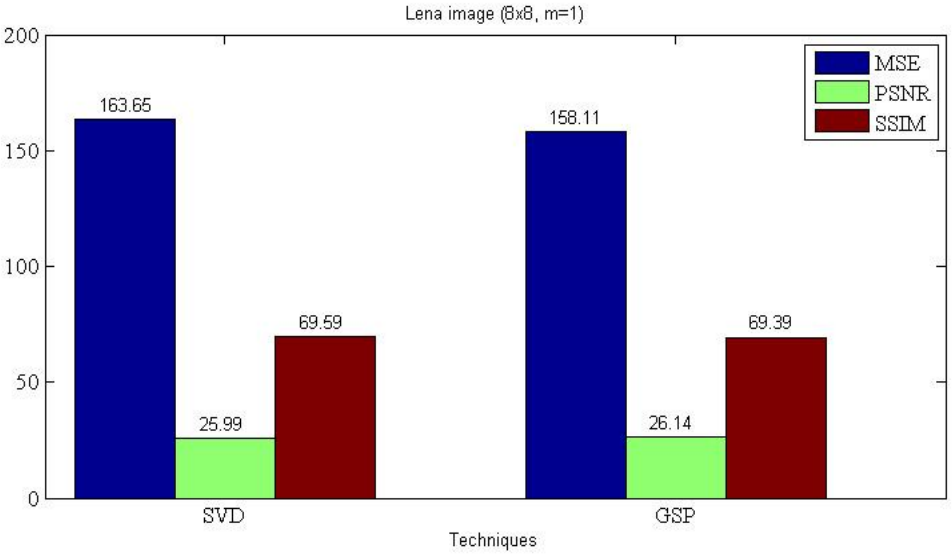


Fig. 5.1. Lena image graph with m=1

(a) (b)

Fig. 5.2. Reconstructed Lena images using (a) SVD and (b) GSP with m=1

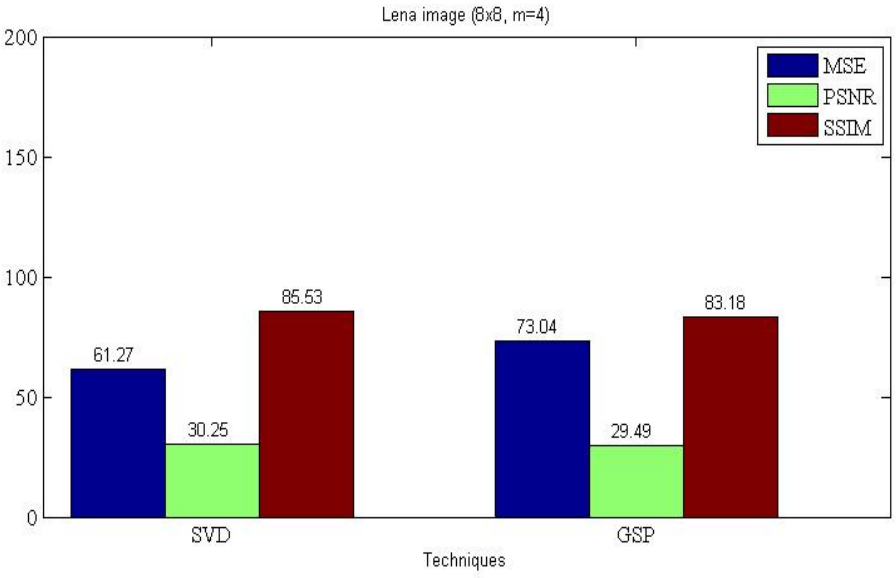


Fig. 5.3. Lena image bar graph with m=4

(a) (b)

Fig. 5.4. Reconstructed Lena images using (a) SVD and (b) GSP with m=4

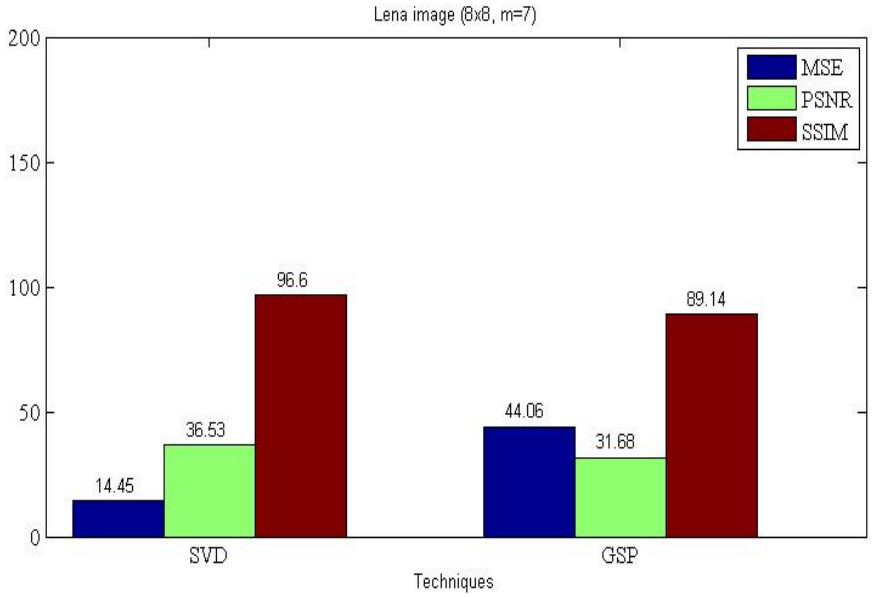


Fig. 5.5. Lena image bar graph with m=7

(a) (b)

Fig. 5.6. Reconstructed Lena images using (a) SVD and (b) GSP with m=7

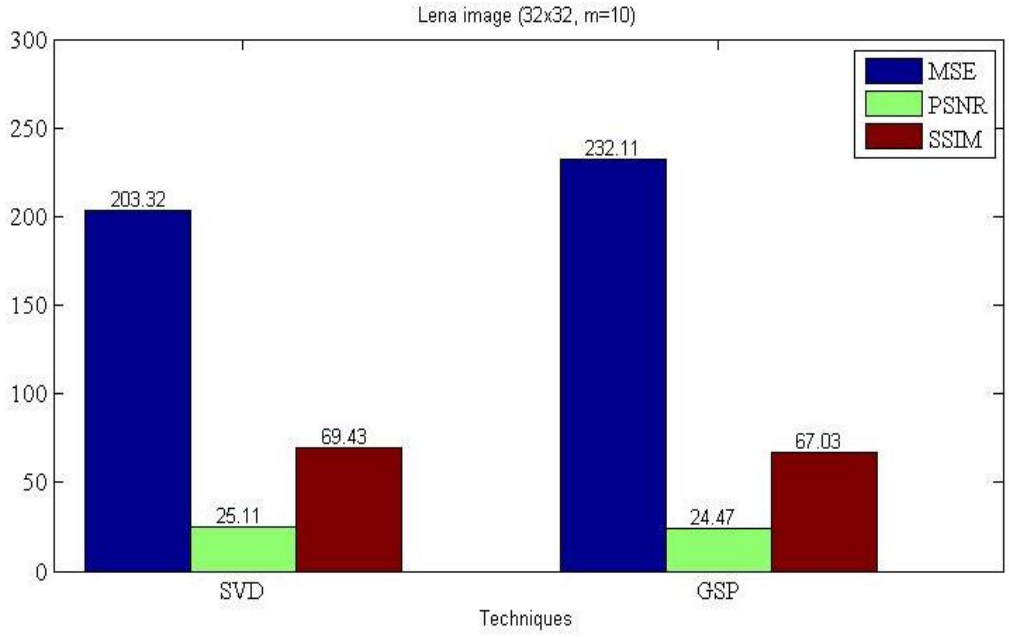


Fig. 5.7. Lena image bar graph with m=10

(a) (b)

Fig. 5.8. Reconstructed Lena images using (a) SVD and (b) GSP with m=10

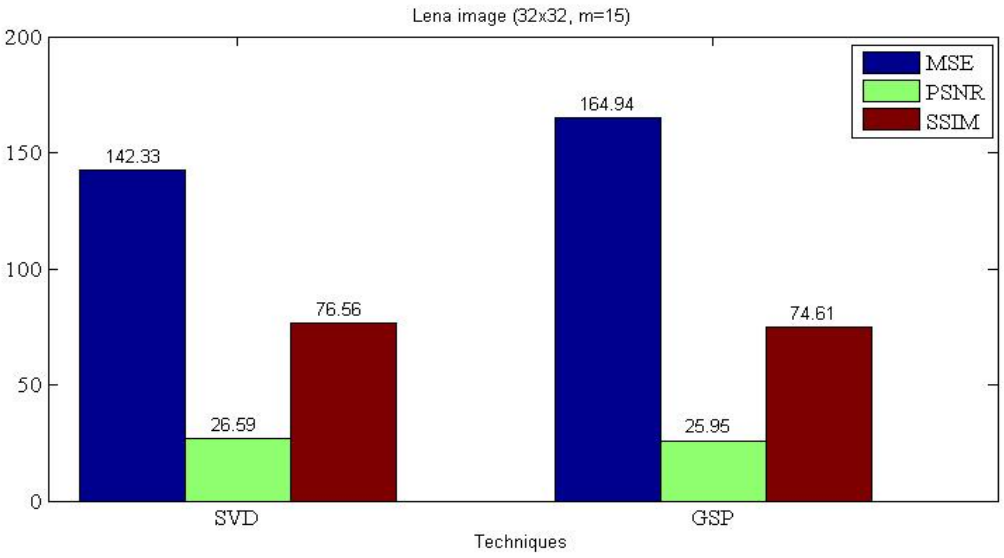


Fig. 5.9. Lena image bar graph with m=15

(a) (b)

Fig. 5.10. Reconstructed Lena images using (a) SVD and (b) GSP with m=15

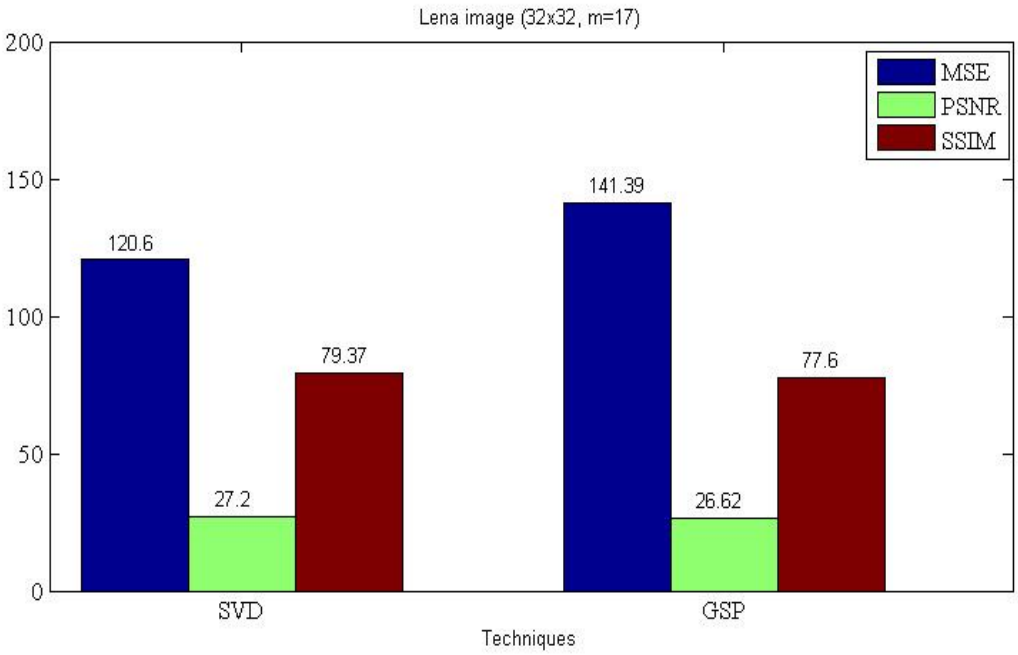


Fig. 5.11. Lena image bar graph with m=17

(a) (b)

Fig. 5.12. Reconstructed Lena images using (a) SVD and (b) GSP with m=17

Table 5.2. Simulated data of reconstructed (Aerial) image with variable block size and principal components

Image	Block size	m=	SVD			GSP		
			MSE	PSNR	SSIM	MSE	PSNR	SSIM
		1	502.29	21.12	55.64	490.01	21.12	56.52
Aerial	8×8	4	249.08	24.16	79.41	262.55	23.93	78.31
		7	57.86	30.51	93.59	57.89	30.49	95.56
		10	442.67	21.67	61.69	502.77	21.11	58.17
Aerial	32×32	15	322.55	23.04	71.79	360.94	22.56	69.29
		17	280.11	23.65	75.26	310.54	23.19	73.23

Table 5.2, represents the simulated data for Aerial image. It represents the data in terms of MSE, PSNR and SSIM with respect of variable block size (8×8, 32×32) and principal components (1, 4, 7; 10, 15, 17) respectively. It is analyzed from the table, for the 8×8 and 32×32 that SVD method has an average gain of 0.7% in SSIM, 0.33% in PSNR and -0.15% in MSE over GSP and 4.01% in SSIM, 2.24% in PSNR and -10.97% in MSE over GSP respective block size.

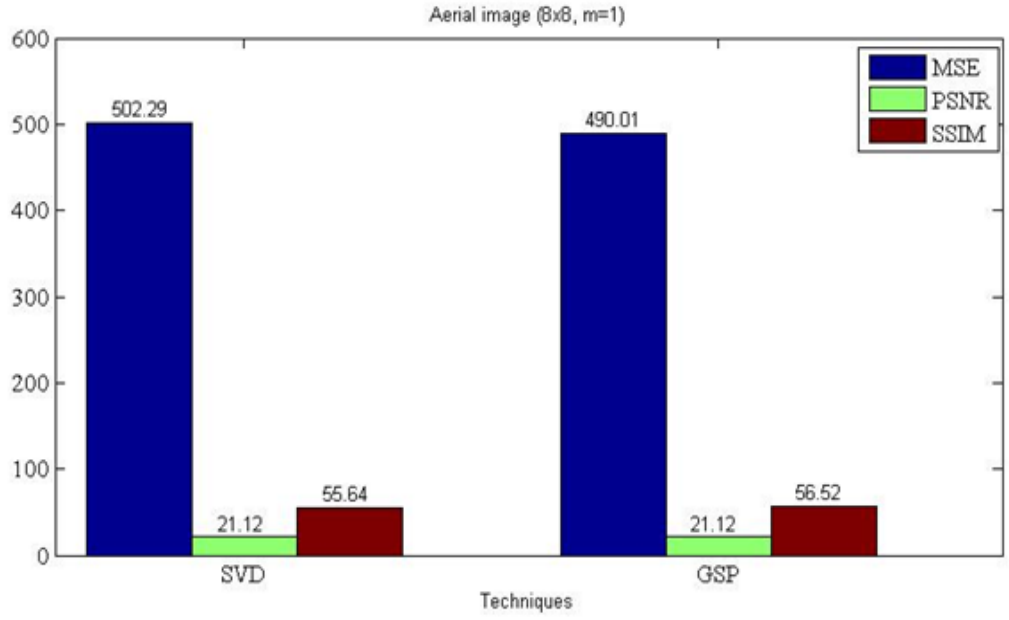


Fig. 5.13. Aerial image bar graph with m=1

(a) (b)

Fig. 5.14. Reconstructed Aerial images using (a) SVD and (b) GSP with m=1

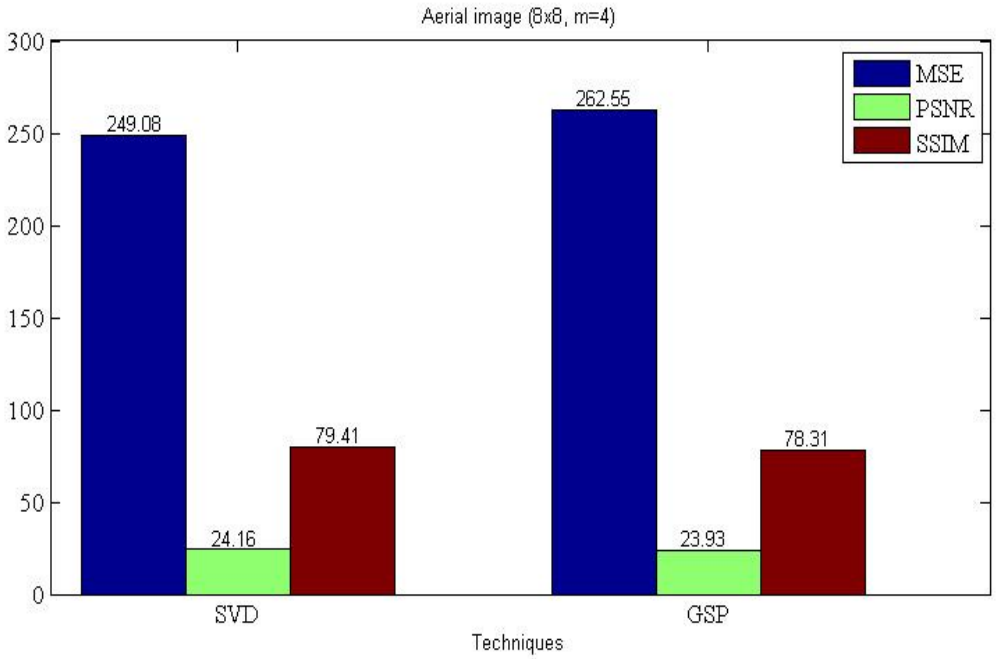


Fig. 5.15. Aerial image bar graph with m=4

(a) (b)

Fig. 5. 16. Reconstructed Aerial images using (a) SVD and (b) GSP with m=4

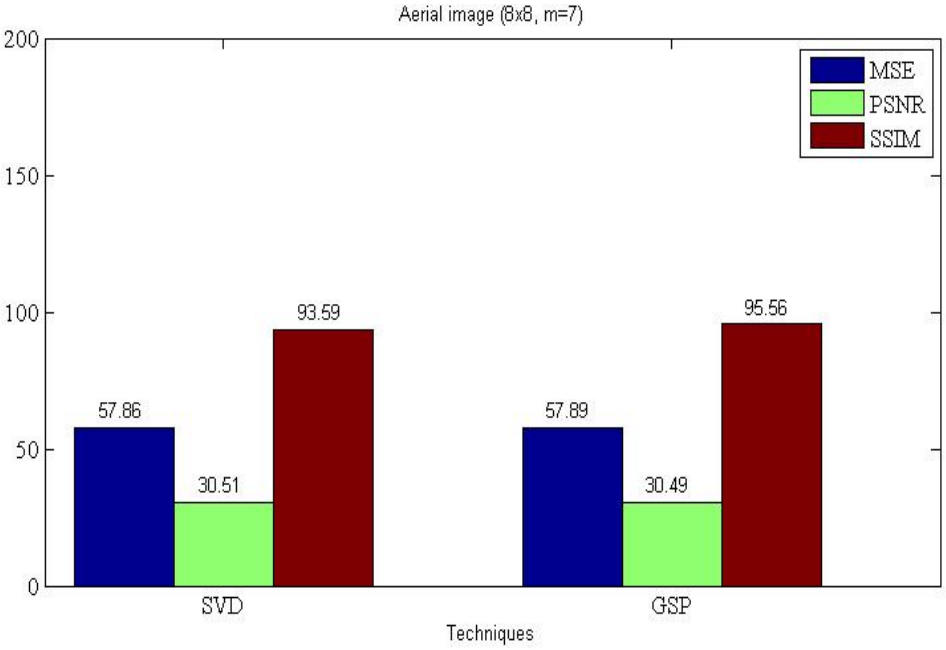


Fig. 5. 17. Aerial image bar graph with m=7

(a) (b)

Fig. 5. 18. Reconstructed Aerial images using (a) SVD and (b) GSP with m=7

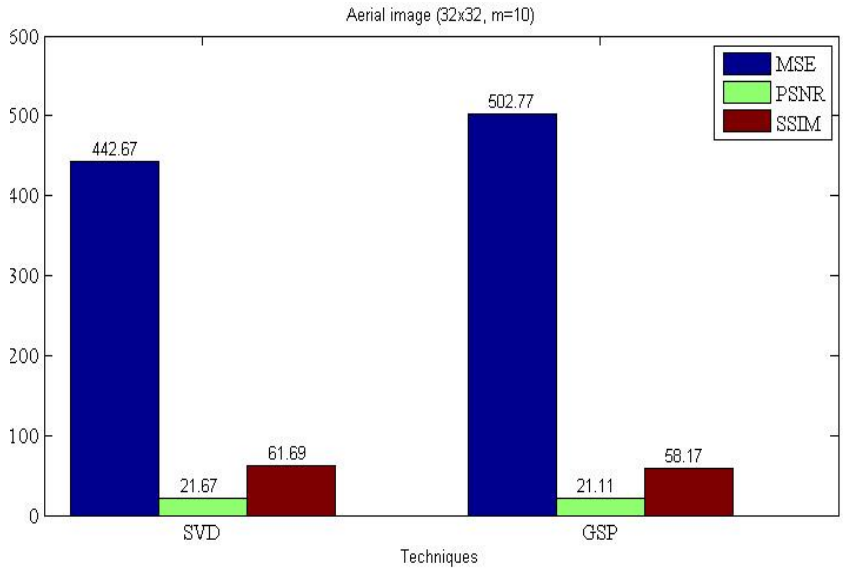


Fig. 5. 19. Aerial image bar graph with m=10

(a) (b)

Fig. 5. 20. Reconstructed Aerial images using (a) SVD and (b) GSP with m=10

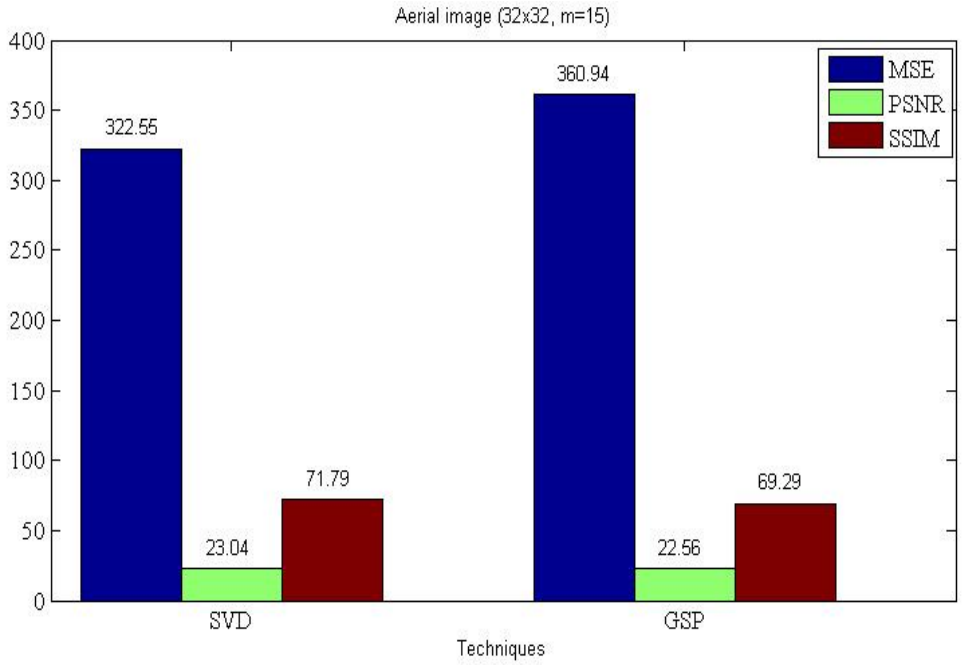


Fig. 5. 21. Aerial image bar graph with m=15

(a) (b)

Fig. 5. 22. Reconstructed Aerial images using (a) SVD and (b) GSP with m=15

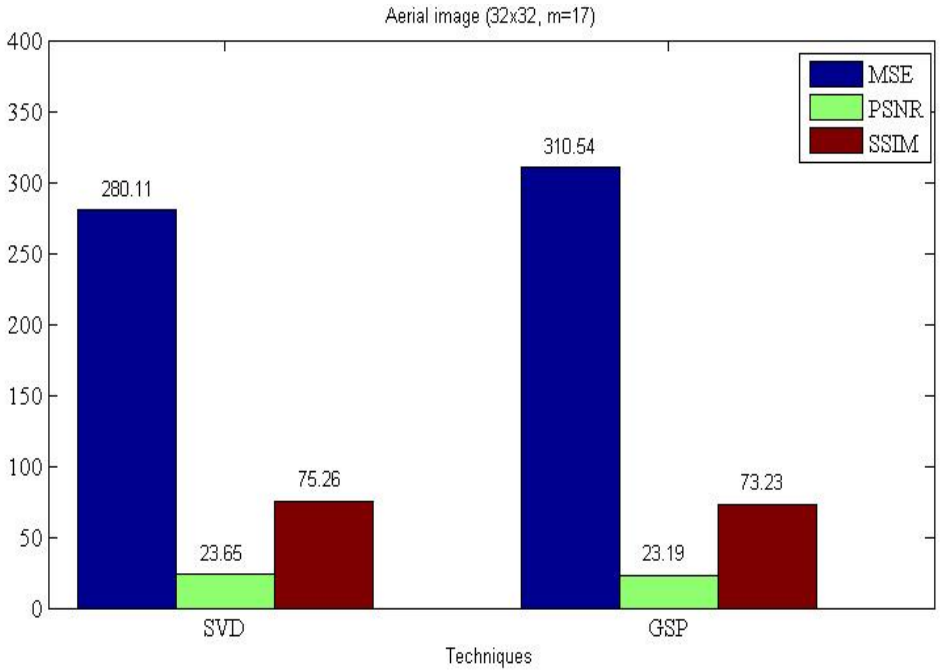


Fig. 5. 23. Aerial image bar graph with m=17

(a) (b)

Fig. 5. 24. Reconstructed Aerial images using (a) SVD and (b) GSP with m=17

Table 5.3. Simulated data of reconstructed (Barbara) image with variable block size and principal components

Image	Block size	m=	SVD			GSP		
			MSE	PSNR	SSIM	MSE	PSNR	SSIM
		1	289.85	23.50	70.09	286.17	23.56	70.62
Barbara	8×8	4	142.83	26.58	85.73	143.28	26.56	86.22
		7	34.23	32.78	96.83	28.64	33.56	95.39
		10	293.33	23.40	69.66	314.26	23.15	68.17
Barbara	32×32	15	208.33	24.94	77.44	223.20	24.64	76.33
		17	181.51	25.48	80.10	190.75	25.32	79.67

Table 5.3, represents the simulated data for Barbara image. It represents the data in terms of MSE, PSNR and SSIM with respect of variable block size (8×8, 32×32) and principal components (1, 4, 7; 10, 15, 17) respectively. It is analyzed from the table, for the 8×8 and 32×32 that SVD method has an average gain of 0.16% in SSIM, 0.99% in PSNR and -1.92% in MSE over GSP and 1.35% in SSIM, 0.97% in PSNR and -6.18% in MSE over GSP respective block size

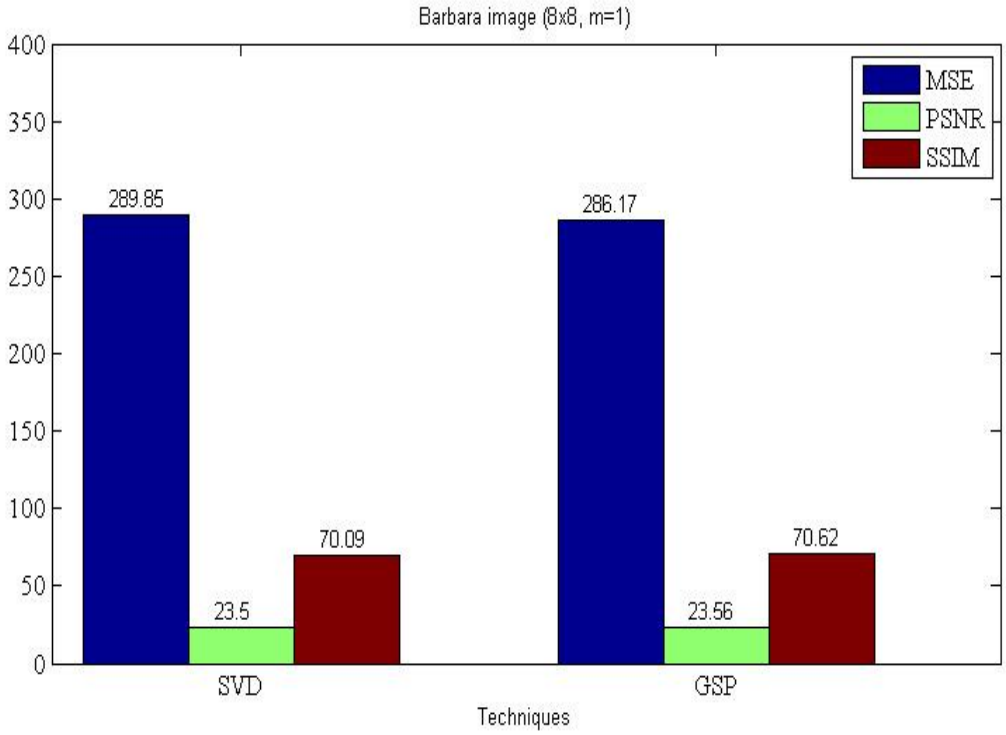


Fig. 5. 25. Barbara image graph with m=1

(a) (b)

Fig. 5. 26. Reconstructed Barbara images using (a) SVD and (b) GSP with m=1

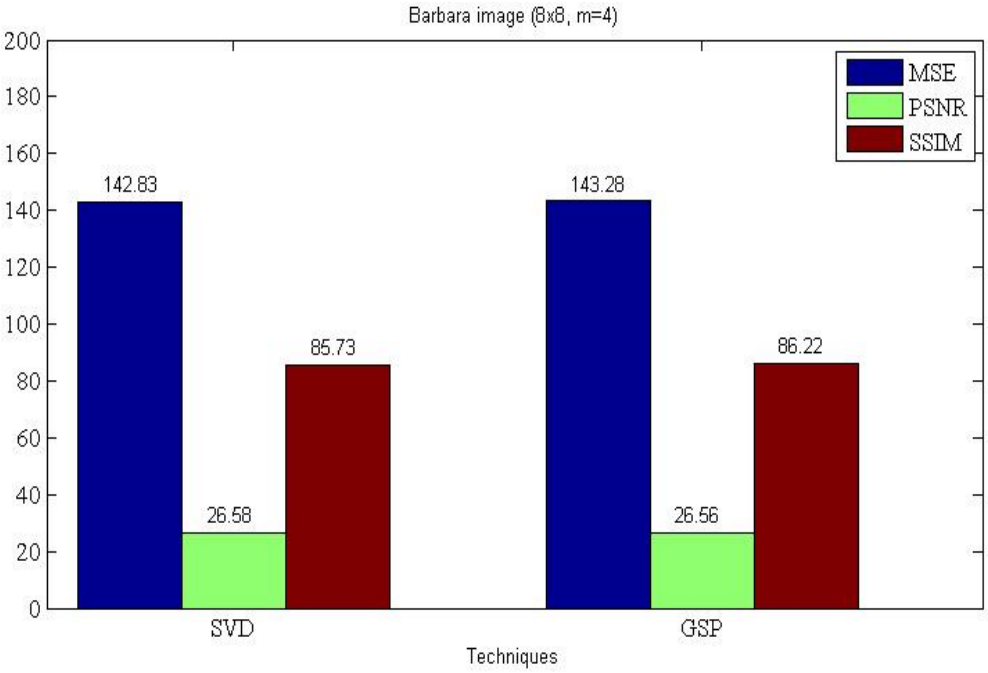


Fig. 5. 27. Barbara image bar graph with m=4

(a) (b)

Fig. 5. 28. Reconstructed Barbara images using (a) SVD and (b) GSP with m=4

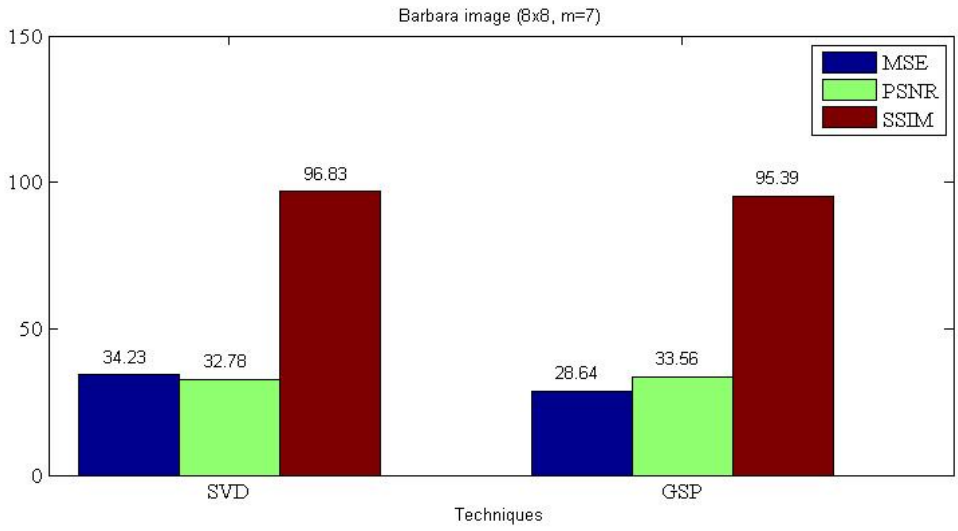


Fig. 5. 29. Barbara image bar graph with m=7

(a) (b)

Fig. 5. 30. Reconstructed Barbara images using (a) SVD and (b) GSP with m=7

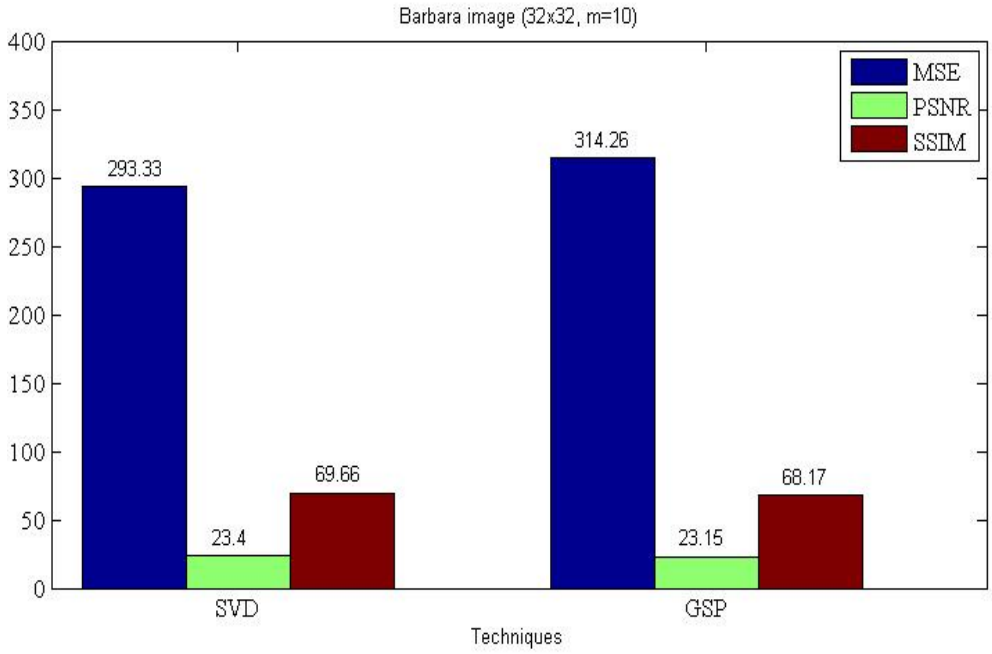


Fig. 5. 31. Barbara image bar graph with m=10

(a) (b)

Fig. 5. 32. Reconstructed Barbara images using (a) SVD and (b) GSP with m=10

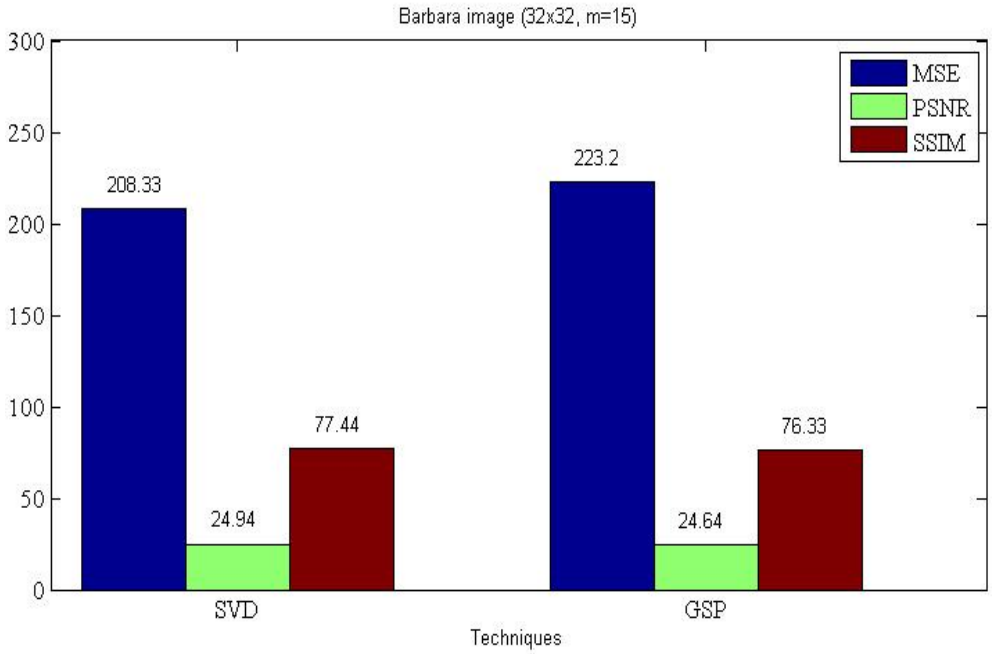


Fig. 5. 33. Barbara image bar graph with m=15

(a) (b)

Fig. 5. 34. Reconstructed Barbara images using (a) SVD and (b) GSP with m=15

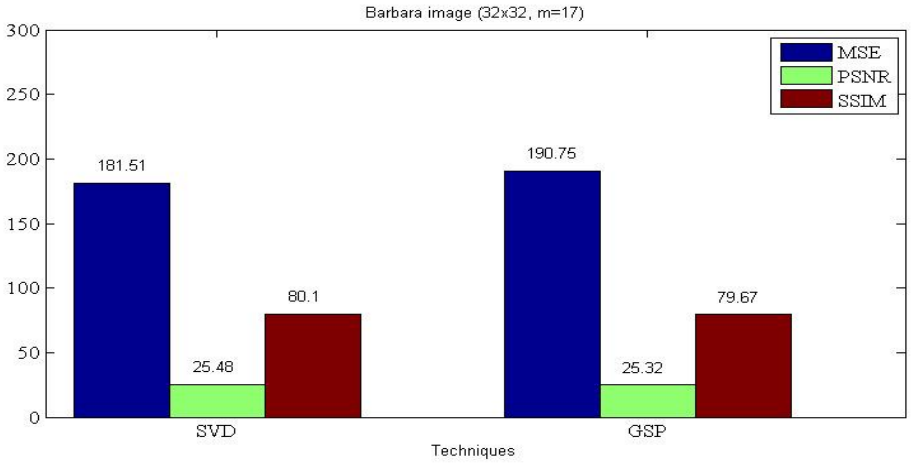


Fig. 5. 35. Barbara image bar graph with m=17

(a) (b)

Fig. 5. 36. Reconstructed Barbara images using (a) SVD and (b) GSP with m=17

Table 5.4. Simulated data of reconstructed (Pepper) image with variable block size and principal components

Image	Block size	m=	SVD			GSP		
			MSE	PSNR	SSIM	MSE	PSNR	SSIM
		1	131.30	26.94	85.09	198.35	25.15	83.09
Pepper	8×8	4	13.99	36.49	96.80	63.43	30.09	93.55
		7	2.46	44.21	99.29	52.07	30.96	96.75
		10	54.04	30.80	89.10	138.01	26.72	80.02
Pepper	32×32	15	22.88	34.53	93.30	77.83	29.21	86.35
		17	17.71	35.64	94.43	85.65	28.79	86.59

Table 5.4, represents the simulated data for Pepper image. It represents the data in terms of MSE, PSNR and SSIM with respect of variable block size ($8\times8, 32\times32$) and principal components (1, 4, 7; 10, 15, 17) respectively. It is analyzed from the table, for the 8×8 and 32×32 that SVD method has an average gain of 2.84% in SSIM, 24.87% in PSNR and -52.92% in MSE over GSP and 9.43% in SSIM, 19.18% in PSNR and -68.16% in MSE over GSP respective block size

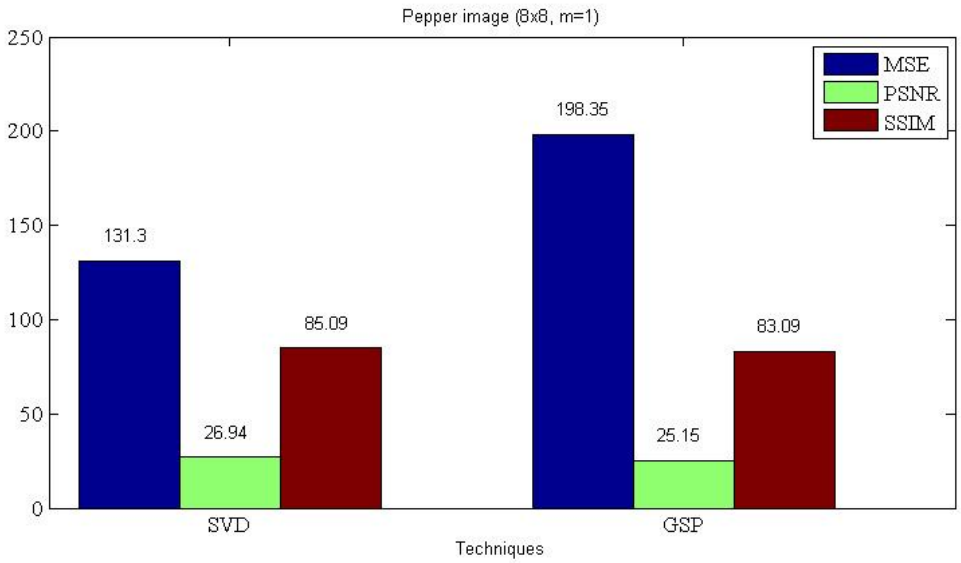


Fig. 5. 37. Pepper image bar graph with m=1

(a) (b)

Fig. 5. 38. Reconstructed Pepper images using (a) SVD and (b) GSP with m=1

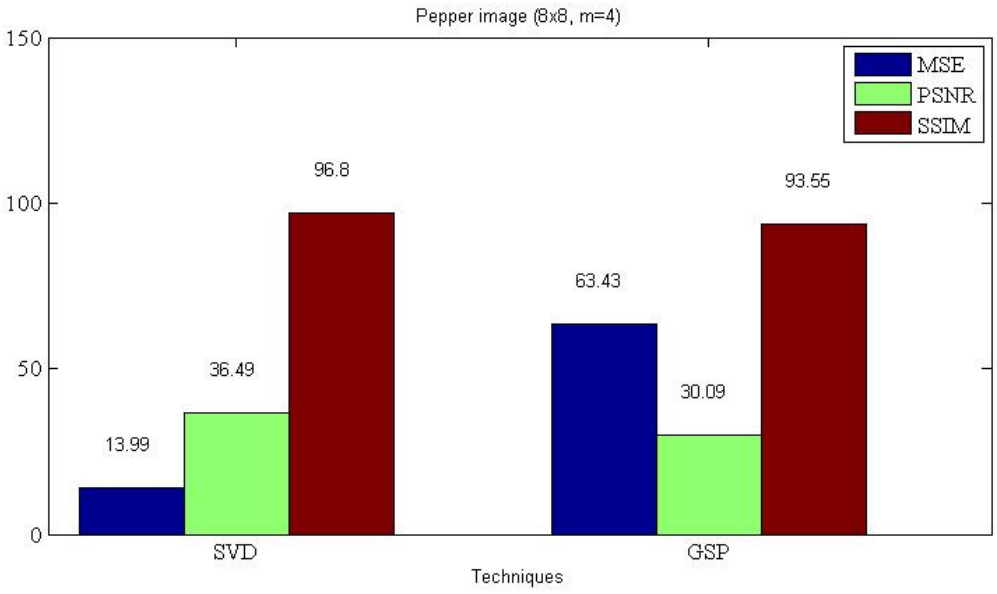


Fig. 5. 39. Pepper image bar graph with m=4

(a) (b)

Fig. 5. 40. Reconstructed Pepper images using (a) SVD and (b) GSP with m=4

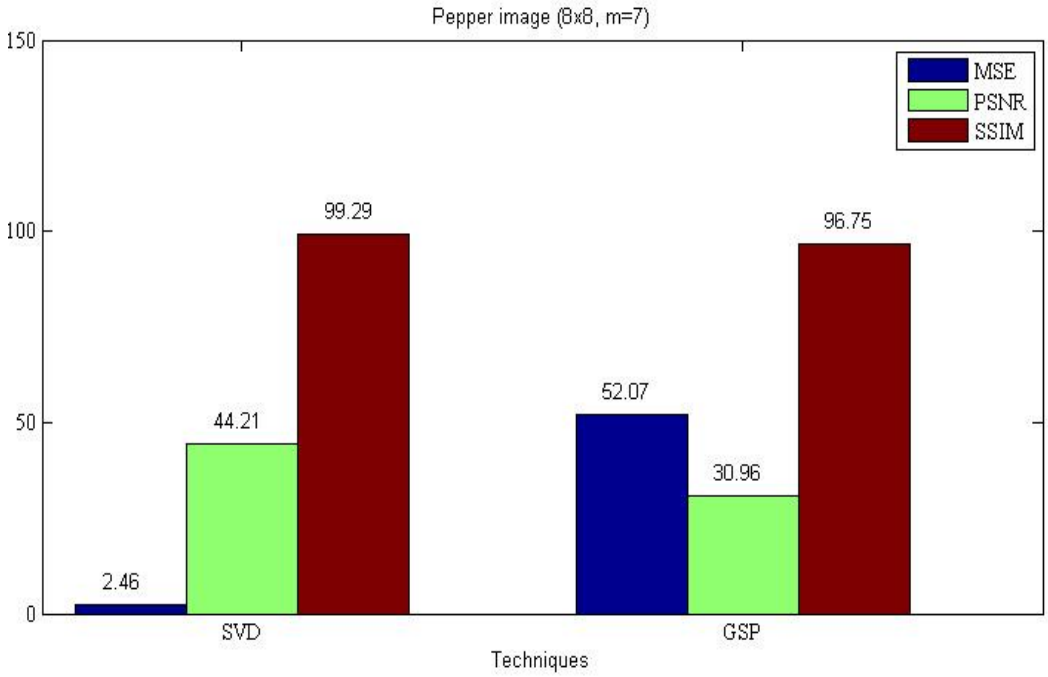


Fig. 5. 41. Pepper image bar graph with m=7

(a) (b)

Fig. 5. 42. Reconstructed Pepper images using (a) SVD and (b) GSP with m=7

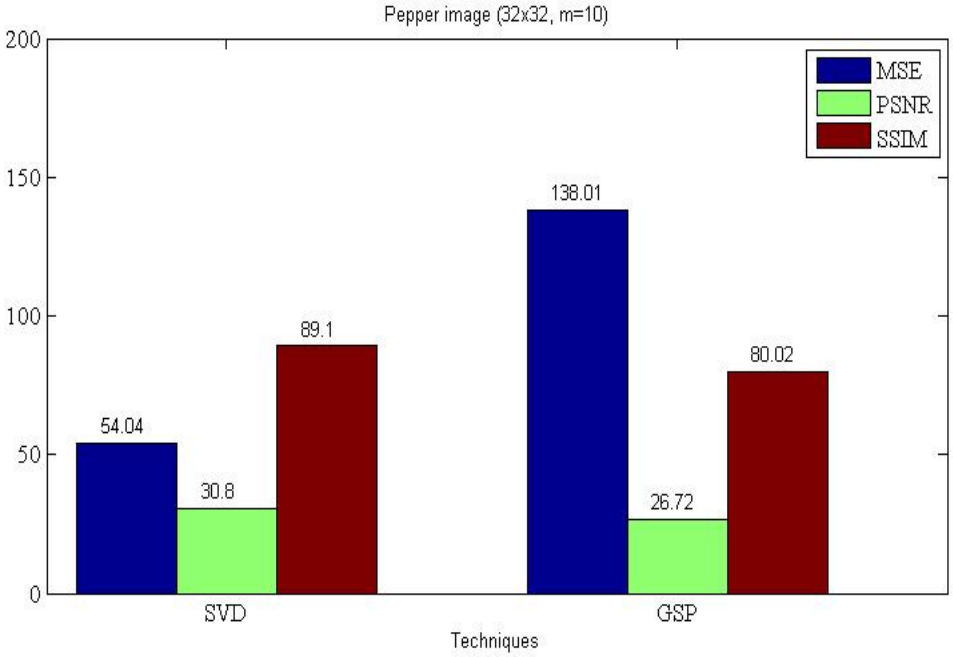


Fig. 5. 43. Pepper image bar graph with m=10

(a) (b)

Fig. 5. 44. Reconstructed Pepper images using (a) SVD and (b) GSP with m=10

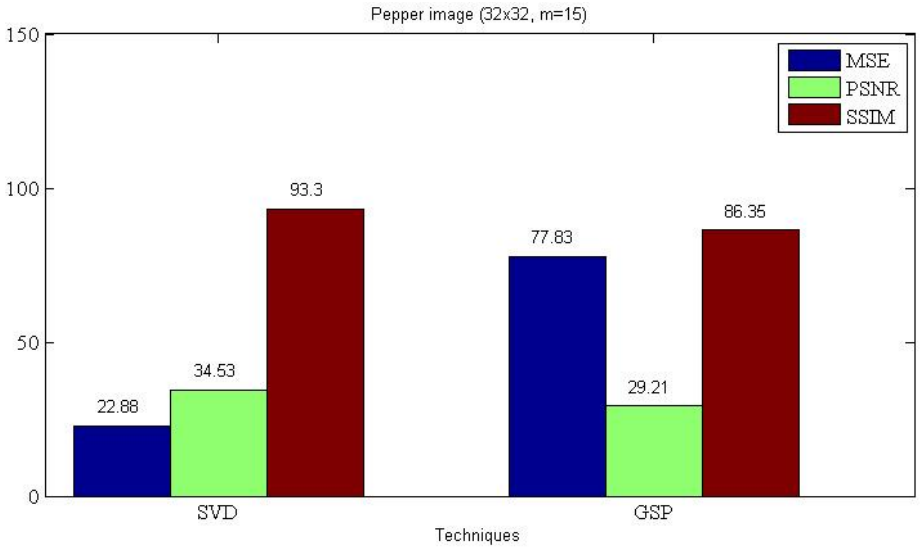


Fig. 5. 45. Pepper image bar graph with m=15

(a) (b)

Fig. 5. 46. Reconstructed Pepper images using (a) SVD and (b) GSP with m=15

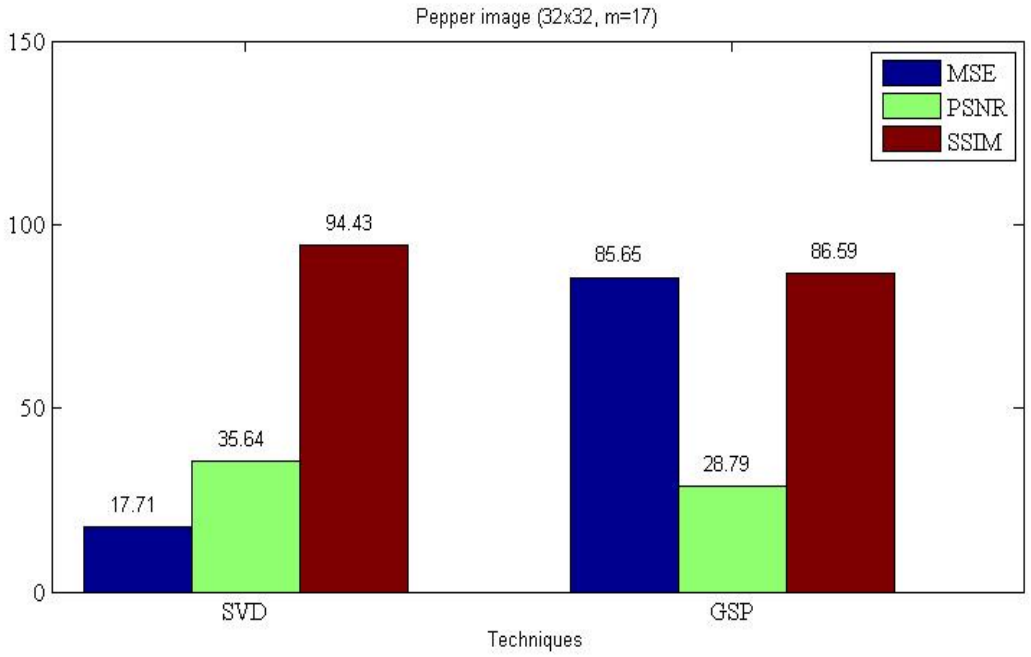


Fig. 5. 47. Pepper image bar graph with m=17

(a) (b)

Fig. 5. 48. Reconstructed Pepper images using (a) SVD and (b) GSP with m=17

Table. 5.5. Simulated data for reconstructed images

Image	Block Size	Principal Component (m =)	Singular Value Decomposition (SVD)			Graham Schmidt Procedure (GSP)		
			MSE	PSNR	SSIM	MSE	PSNR	SSIM
Lena	32x32	28	125.09	27.15	89.34	128.09	27.05	89.09
Aerial	32x32	28	83.89	28.89	91.79	84.02	28.88	91.59
Barbara	32x32	28	74.48	29.41	92.89	80.17	29.09	92.09
Pepper	32x32	28	59.98	30.35	90.72	82.83	28.94	90

Table 5.5, represents the simulated data for images named as Lena, Aerial, Barbara and Pepper. It represents the data in terms of MSE, PSNR and SSIM with respect of 32×32 block of each image with m=28. It is analyzed from the table, for the Lena and Aerial images that SVD method has a gain of 0.28% in SSIM, 0.36% in PSNR and -2.34% in MSE; 0.21% in SSIM, 0.03% in PSNR and -0.51% in MSE over GSP respectively. As far as Barbara and Pepper images are concerned it has a gain of 0.86% in SSIM, 1.1% in PSNR, -7.09% in MSE; 0.8% in SSIM, 4.8% in PSNR and -27.58% in MSE over GSP respectively.

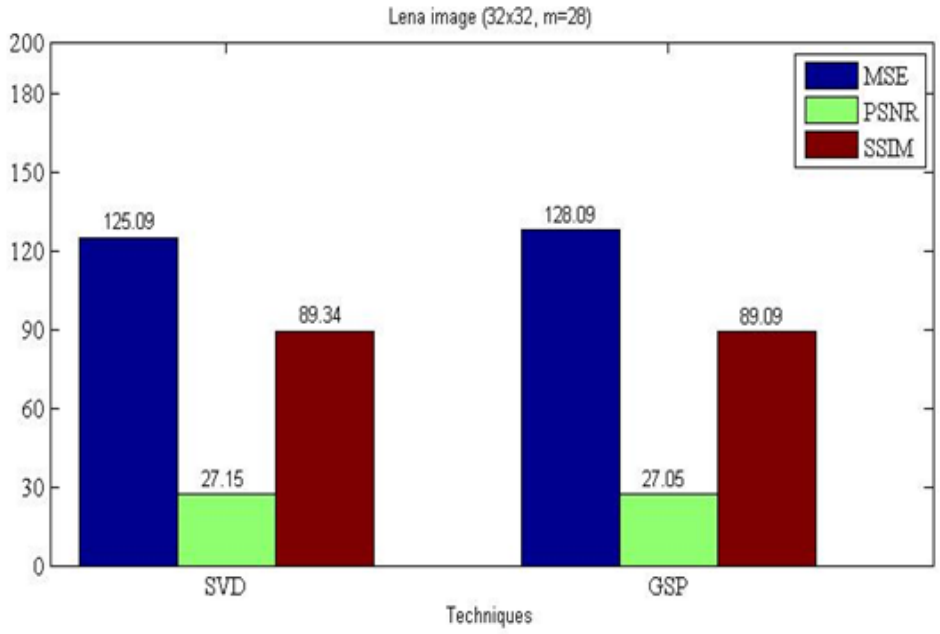


Fig. 5.49. Lena image bar graph

(a) (b)

Fig. 5.50. Reconstructed Lena images using (a) SVD and (b) GSP

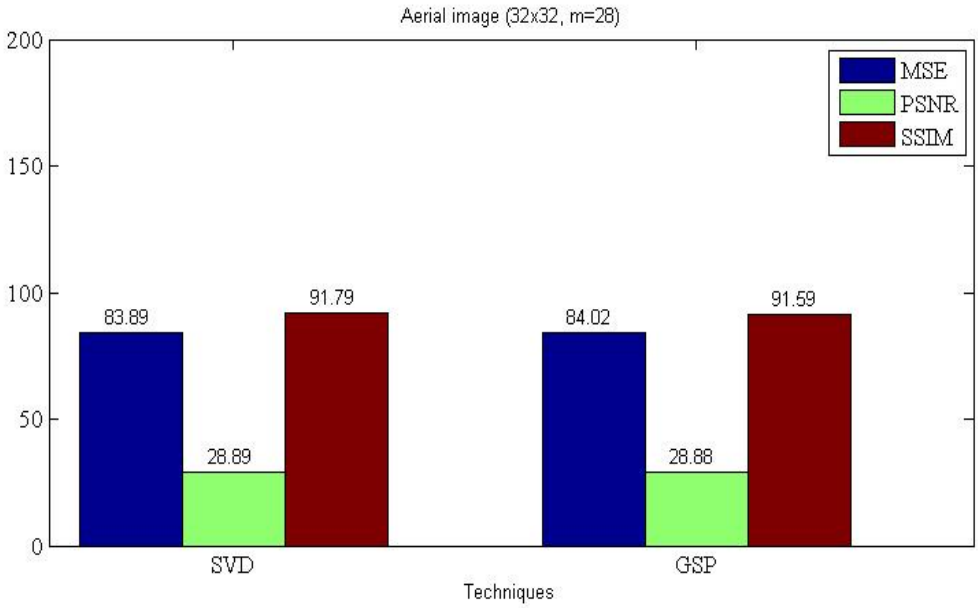


Fig. 5.51. Aerial image bar graph

(a) (b)

Fig. 5.52. Reconstructed Aerial images using (a) SVD and (b) GSP

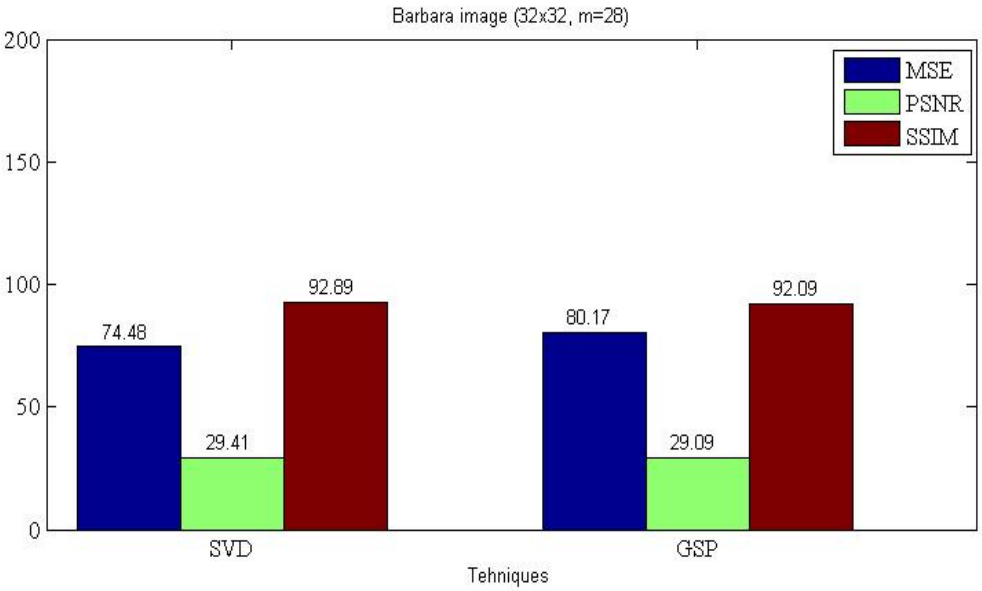


Fig. 5.53. Barbara image bar graph

(a) (b)

Fig. 5.54. Reconstructed Barbara images using (a) SVD and (b) GSP

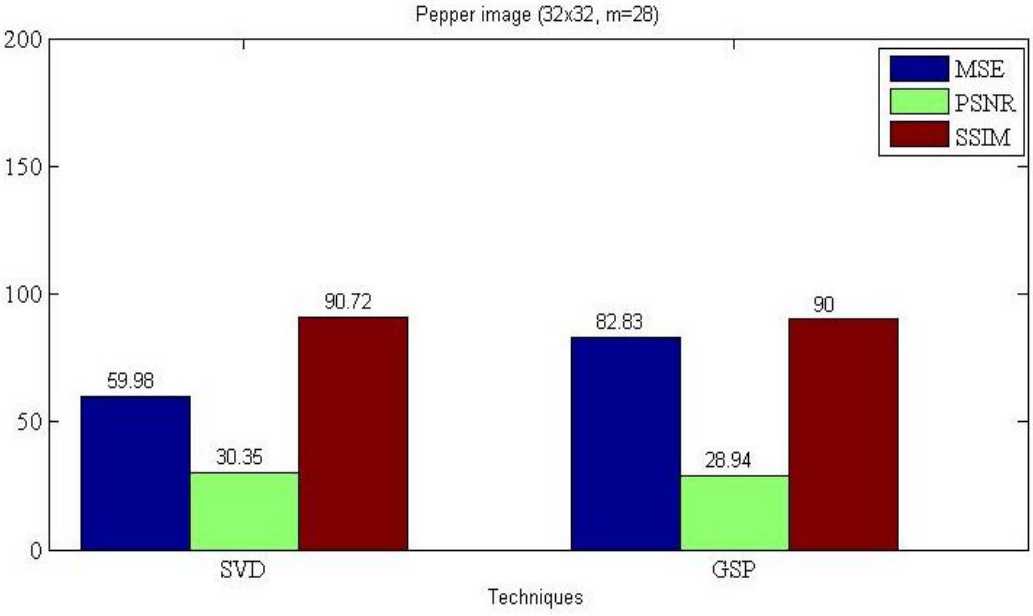


Fig. 5.55. Pepper image bar graph

(a) (b)

Fig. 5.56. Reconstructed Pepper images using (a) SVD and (b) GSP

5.2. GHA Method for PCA with 100 Iterations

In this section, we will represent the image data compression for Generalized Hebbian Algorithm (GHA) which is based on the unsupervised neural training network for 100 iterations. Data tables, bar graphs and reconstructed images are given below, representing the comparison between GHA, SVD and GSP methods.

Table. 5.6. Simulated data for reconstructed images including GHA (100 iterations)

Image	Block Size	m =	SVD			GSP			GHA		
			MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Lena	32x32	28	125.09	27.15	89.34	128.09	27.05	89.09	688.02	19.75	63.66
Aerial	32x32	28	83.89	28.89	91.79	84.02	28.88	91.59	494.59	21.18	57.47
Barbara	32x32	28	74.48	29.41	92.89	80.17	29.09	92.09	363.60	22.52	73.33
Pepper	32x32	28	59.98	30.35	90.72	82.83	28.94	90	404.13	22.06	71.22

Table 5.6, represents a comparison among the SVD, GSP and GHA for Lena, Aerial, Barbara and Pepper images. It also represents the simulated data in terms of MSE, PSNR and SSIM with respect of 32×32 block of each image with m=28. A comparison between the SVD and GSP is analyzed in table 5.1 which justify that SVD method is better than the GSP method. In table 5.2, it is analyzed for the Lena and Aerial images that SVD method has a gain of 40.33% in SSIM, 37.46% in PSNR and -81.81% in MSE; 59.71% in SSIM, 36.40% in PSNR and -83.00% in MSE over GHA respectively. As far

as Barbara and Pepper image is concerned, it has a gain of 9.32% in SSIM, 30.59% in PSNR and -79.51% in MSE; 27.37% in SSIM, 37.57% in PSNR and -85.15% in MSE over GHA respectively. It is clear from the above data that SVD method performs better than GHA method for 100 iterations.

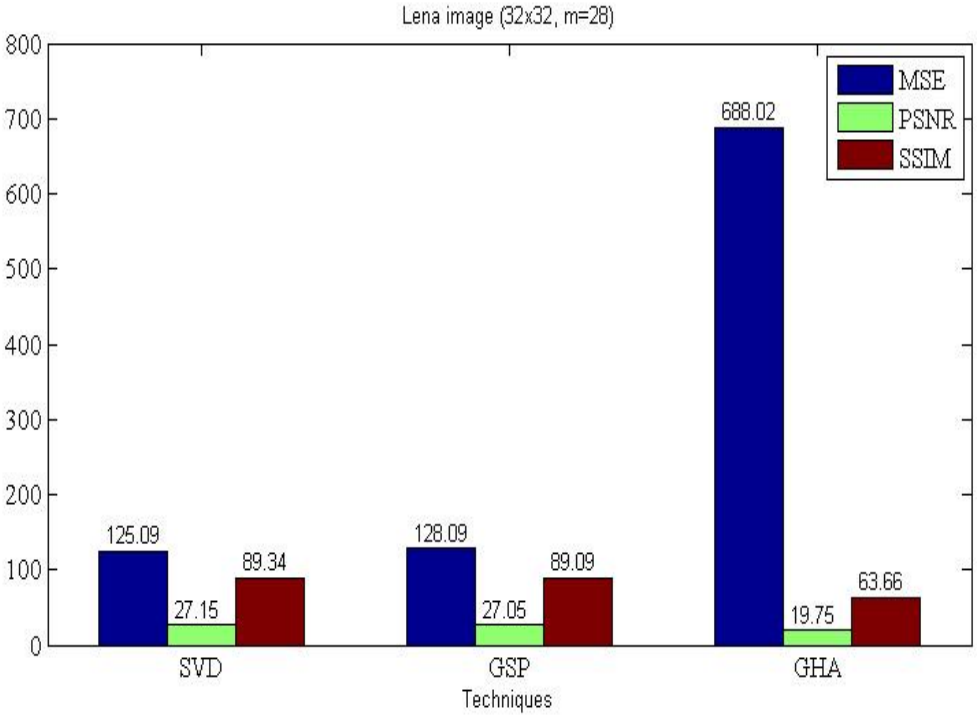


Fig. 5.57. Lena image bar graph with GHA method (100 iterations)

(a) (b) (c)

Fig. 5.58. Reconstructed Lena images using (a) GHA (100 iterations) (b) SVD and (c) GSP

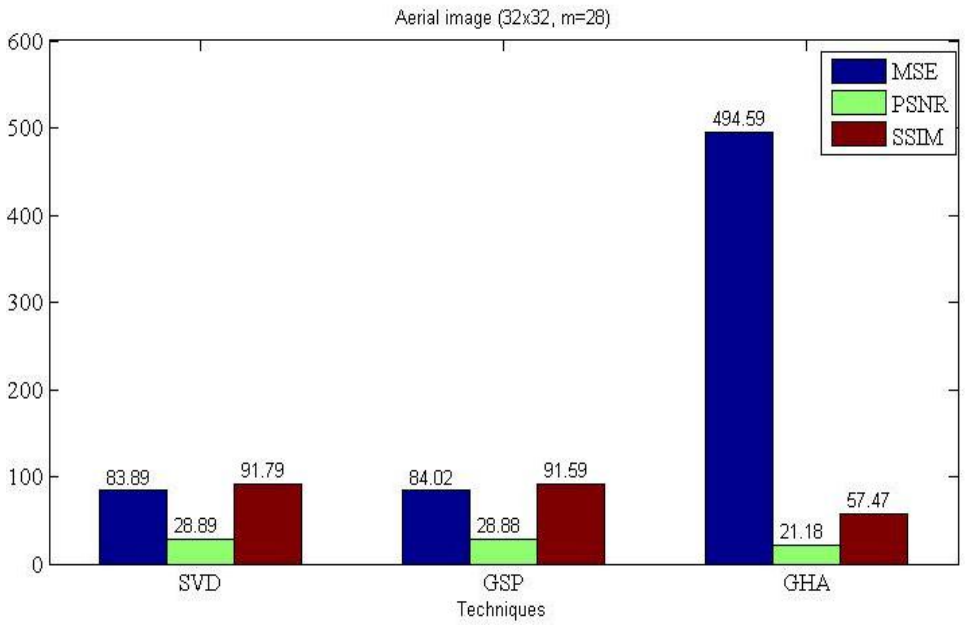


Fig. 5.59. Aerial image bar graph with GHA method (100 iterations)

(a) (b) (c)

Fig. 5.60. Reconstructed Aerial images using (a) GHA (100 iterations) (b) SVD and (c) GSP

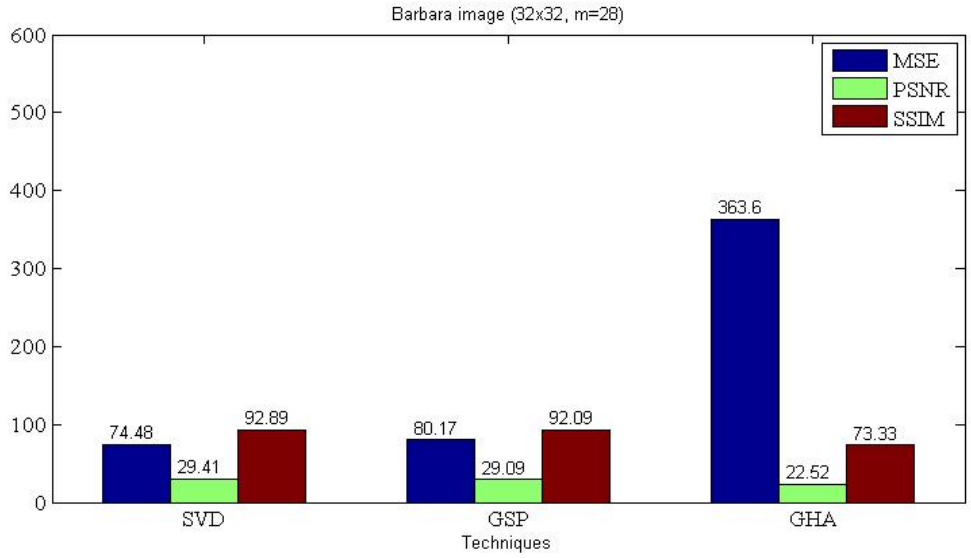


Fig. 5.61. Barbara image bar graph with GHA method (100 iterations)

Fig. 5.62. Reconstructed Barbara images using (a) GHA (100 iterations) (b) SVD and (c) GSP

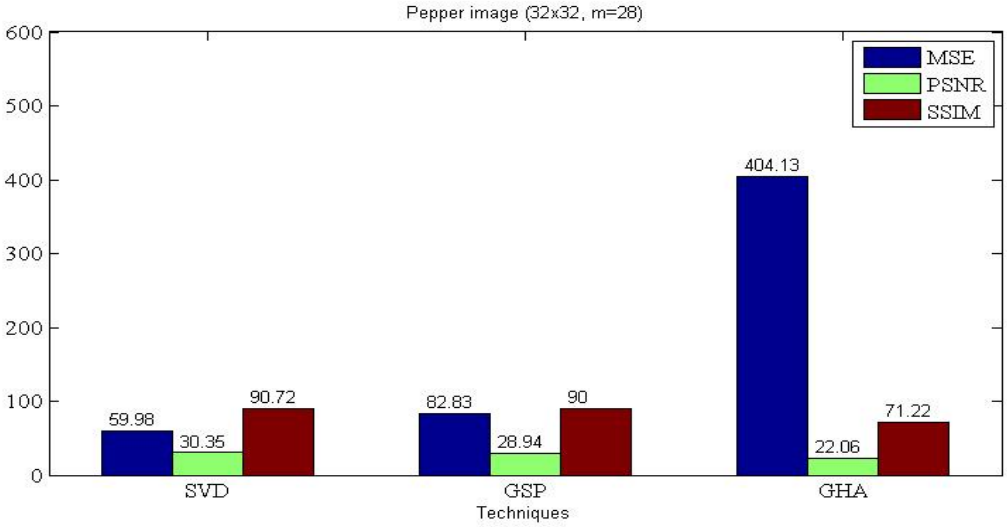


Fig. 5.63. Pepper image bar graph with GHA method (100 iterations)

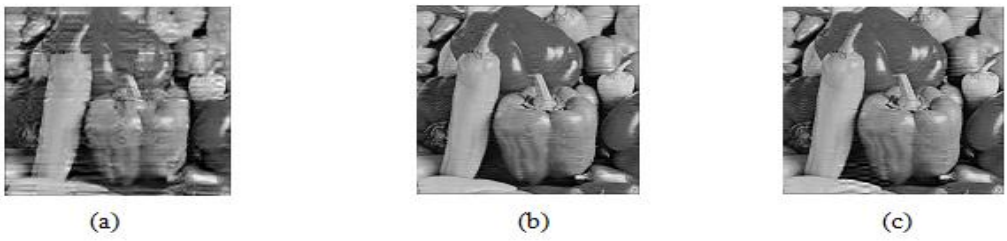


Fig. 5.64. Reconstructed Pepper images using (a) GHA (100 iterations) (b) SVD and (c) GSP

5.3. GHA Method for PCA with 500 Iterations

In this section, we will represent the image data compression for Generalized Hebbian Algorithm (GHA) which is based on the unsupervised neural training network for 500 iterations. Data tables, bar graphs and reconstructed images are given below, representing the comparison between GHA, SVD and GSP methods.

Table. 5.7. Simulated data for reconstructed images including GHA (500 iterations)

Image	Block Size	m =	SVD			GSP			GHA		
			MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Lena	32x32	28	125.09	27.15	89.34	128.09	27.05	89.09	61.67	30.22	93.41
Aerial	32x32	28	83.89	28.89	91.79	84.02	28.88	91.59	68.46	29.77	93.19
Barbara	32x32	28	74.48	29.41	92.89	80.17	29.09	92.09	33.12	32.92	95.63
Pepper	32x32	28	59.98	30.35	90.72	82.83	28.94	90	54.88	30.73	92.01

Table 5.7, represents the simulated data for images named as Lena, Aerial, Barbara and Pepper. It represents the data in terms of MSE, PSNR and SSIM with respect of 32×32 block of each image with m=28. It is systematically analyzed from the table, that GHA method for four images has an average gain of 2.60% in SSIM, 6.77% in PSNR and -36.48% in MSE over SVD and 3.16% in SSIM, 8.49% in PSNR and -41.84% in MSE over GSP respectively.

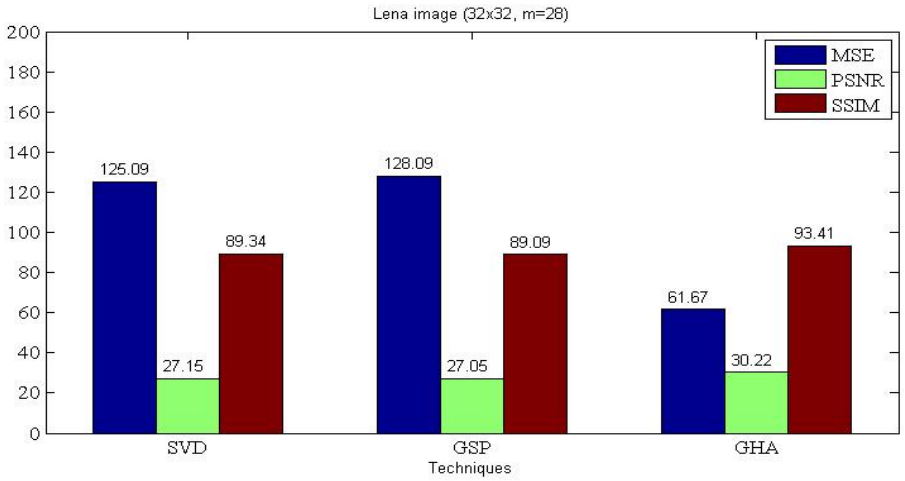


Fig. 5.65. Lena image bar graph with GHA method (500 iterations)

Fig. 5.66. Reconstructed Lena images using (a) GHA (500 iterations) (b) SVD and (c) GSP

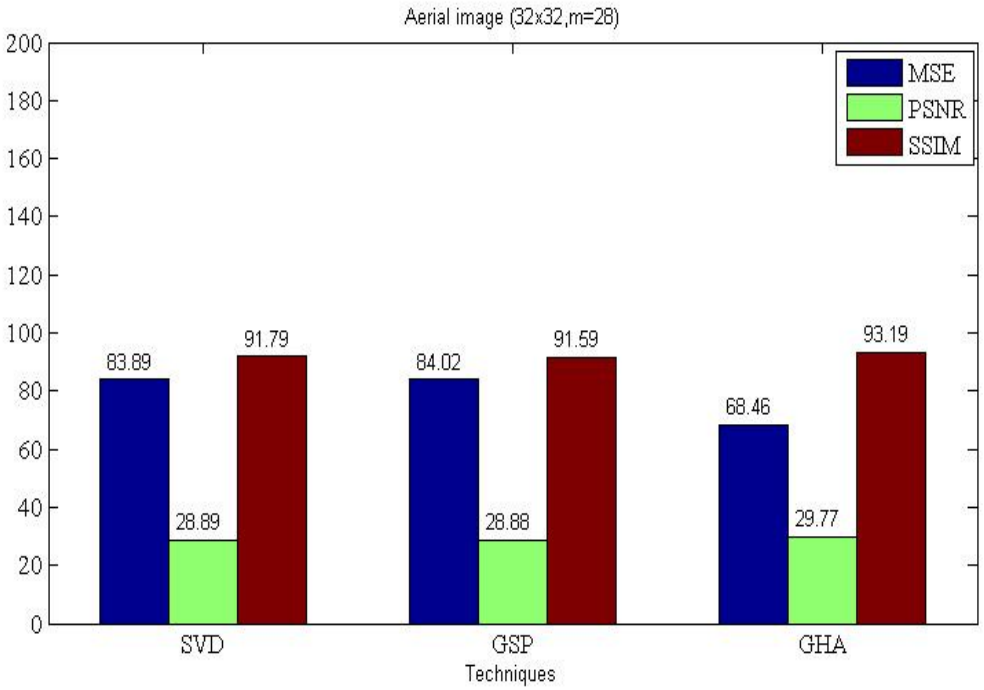


Fig. 5.67. Aerial image bar graph with GHA method (500 iterations)

Fig. 5.68. Reconstructed Aerial images using (a) GHA (500 iterations)
(b) SVD and (c) GSP

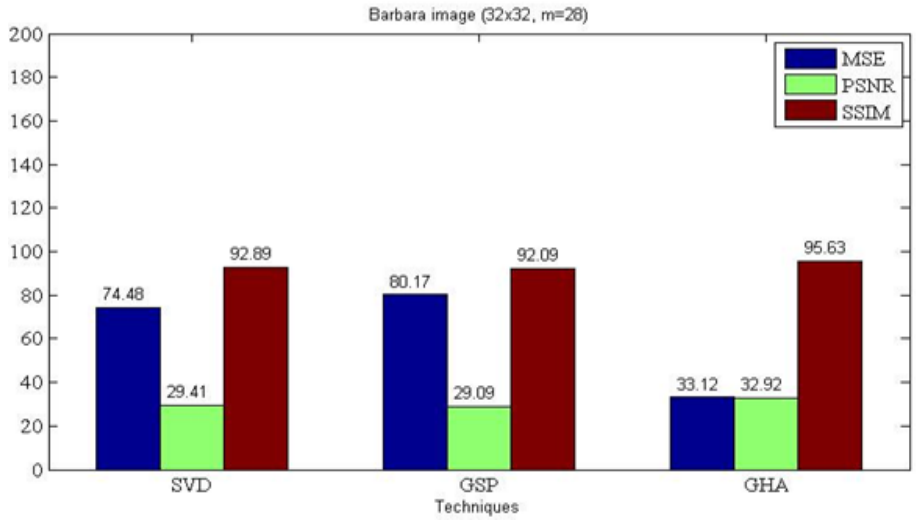
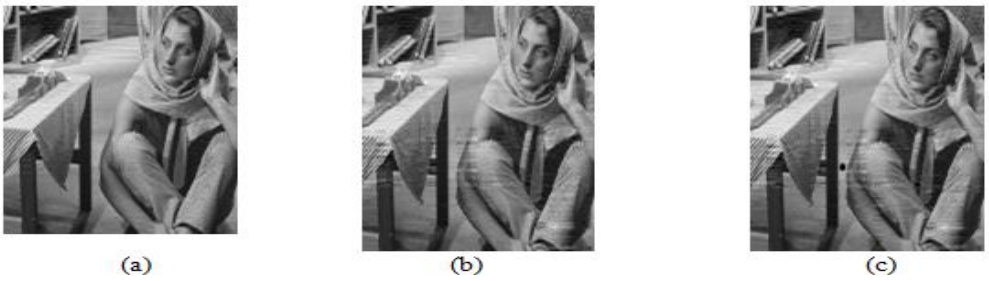


Fig. 5.69. Barbara image bar graph with GHA method (500 iterations)



(a) (b) (c)

Fig. 5.70. Reconstructed Barbara image using (a) GHA (500 iterations)

(b) SVD and (c) GSP

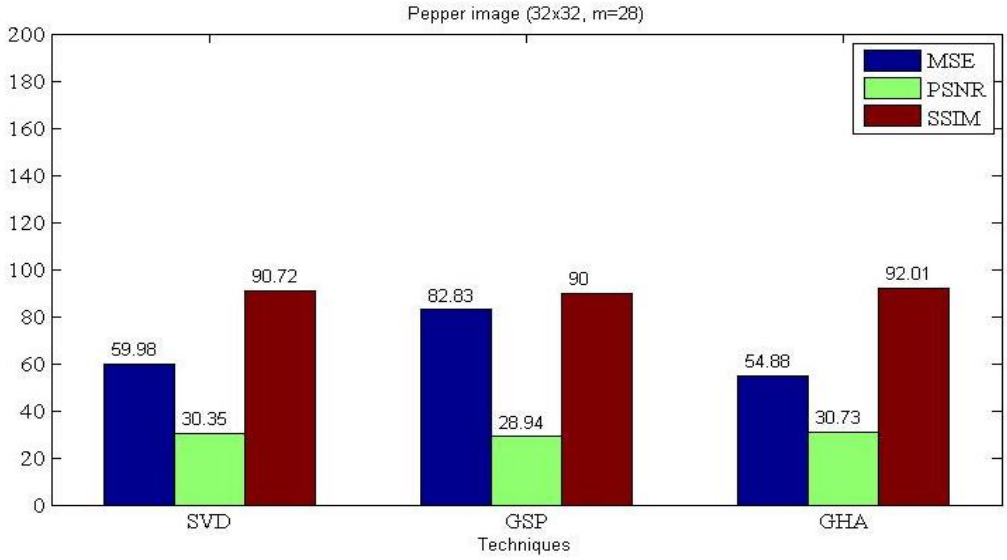


Fig. 5.71. Pepper image bar graph with GHA method (500 iterations)

(a) (b) (c)

Fig. 5.72. Reconstructed Pepper image using (a) GHA (500 iterations)

(b) SVD and (c) GSP

Chapter 6

CONCLUDING REMARKS AND FUTURE SCOPE

6.1. Concluding Remarks

Three different image compression methods named as Singular Value Decomposition, Gram Schmidt Procedure and Generalized Hebbian Algorithm based on principal components have been analyzed. These methods are performed on four different types of images. Their performance is evaluated based on three image quality parameters known as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Structure Similarity Index (SSIM). In this work, we make a comparison between two traditional image compression methods named as SVD and GSP. After analyzing the simulated data given in Table 5.1, 5.2, 5.3 and 5.4 of different block size of an image and different number of extracted principal components, it is evaluated that SVD method is better than GSP in terms of MSE, PSNR and SSIM.

Then, we discuss another method of image compression based on an iterative neural network technique known as GHA. This method is also performed on the same data set of images. After carefully analyzing Table 5.6 and 5.7, it is concluded that the performance of GHA method becomes better as the number of iterations increases, for n=100, its performance is lower as compared to two traditional techniques; but for n=500, it significantly performs better out of the two methods.

It is analyzed from Table 5.7, for Lena image, GHA method has a gain of 4.5% in SSIM, 11.34% in PSNR and -50% in MSE over SVD; 4.8% in SSIM, 11.71% in PSNR and -51.85% in MSE over GSP respectively. As far as the Aerial image is concerned, it has a gain of 1.5% in SSIM, 3.04% in PSNR and -18.39% in MSE over SVD; 1.7% in SSIM, 3.04% in PSNR and -18.51% in MSE over GSP respectively, for the Barbara image, GHA method has a gain 2.94% in SSIM, 11.93% in PSNR and -55.53% in MSE over SVD; 3.84% in SSIM, 13.16% in PSNR and -58.68% in MSE over GSP respectively and Pepper image has a gain of 1.42% in SSIM, 1.25% in PSNR and -8.50% in MSE over SVD; 2.23% in SSIM, 6.18% in PSNR and -33.74% in MSE over GSP respectively.

6.2. Future Scope

The convergence time for the updation of the synaptic weights which served as the principal components for the image data compression is very slow.

REFRENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2^{nd} ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[2] A. N. Netravali and B. G. Haskell, *Digital pictures representation and compression*, 2^{nd} ed. Spring Street, NY: Plenum, 1989.

[3] T. A. Laskar and K. Hemachandran, "An approach for color image compression of JPEG and PNG images using DCT and DWT," *Int. Conf. on Computational Intell. and Commun. Networks*, pp. 129-133, Nov. 2014.

[4] N. Sharma, I. Garg, P. K. Garg, and D. Sharma, "Analysis of transform techniques for 2D compression," *Int. J. of Eng. and Innovative Technol.*, vol. 1, no. 5, pp. 78-82, May 2012.

[5] Z. Liu and L. J. Karam, "Mutual information based analysis of JPEG2000 contexts," *IEEE Trans. on Image Process.*, vol. 14, no. 4, pp. 411-422, Apr. 2005.

[6] Z. Xiong, O. G. Guleryuz, and M. T. Orchard, "A DCT based embedded image coder," *IEEE Signal Process. Lett.*, vol. 3, no. 11, pp. 289-290, Nov. 1996.

[7] S. Miaou, F. Sheng, and S. Chen, "A lossless compression method for medical image sequences using JPEG-LS and interframe coding," *IEEE Trans. on Inform. Technol. in Biomedicine*, vol. 13, no. 5, pp. 818-821, Sept. 2009.

[8] G. K. Wallance, "The JPEG picture compression standard," *IEEE Trans. on Consum. Electron.*, vol. 38, no. 1, pp. 108-124, Feb. 1992.

[9] Y. Ye and P. Cosman, "Dictionary design for text image compression with JBIG2," *IEEE Trans. on Image Process.*, vol. 10, no. 6, pp. 818-828, Jun. 2001.

[10] A. K. Jain, "Image Data Compression: A Review," *Proc. IEEE*, vol. 69, no. 3, pp. 349-389, Mar. 1981.

[11] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feed forward neural network," *Neural Networks*, vol. 2, no. 6, pp. 459-473, Apr. 1989.

[12] E. Oja, "Principal components, Minor components and Linear neural networks," *Neural Networks*, vol. 5, no. 6, pp. 927-935, Dec. 1992.

[13] H. M. Abbas and M. M. Fahmy, "Neural model for Karhunen-Loeve transform with application to adaptive image compression," *Proc. IEE.*, vol. 140, no. 2, pp. 135-143, Apr. 1993.

[14] H. M. Abbas and M. M. Bayoumi, "Anti-Hebbian rule for faster back propagation learning," *IEEE Int. Conf. on Neural Networks*, vol. 1, pp. 101-106, Jul. 1994.

[15] R. D. Dony and S. Haykin, "Neural network approaches to image compression," *Proc. IEEE*, vol. 83, no. 2, pp. 288-303, Feb. 1995.

[16] A. Namphol, S. H. Chin, and M. Arozullah, "Image compression using hierarchical neural network," *IEEE Trans. Aerosp. and Electron. Syst.*, vol. 32, no. 1, pp. 326-338, Jan. 1996.

[17] O. A. Wahhab and M. M. Fahmy, "Image compression using multilayer neural networks," *Proc. IEE*, vol. 144, no. 5, pp. 307-312, Oct. 1997.

[18] C. Clausen and H. Wechasler, "Color image compression using PCA and backpropagation learning," *Pattern Recognition*, vol. 33, no. 9, pp. 1555-1560, Sept. 2000.

[19] S. Costa and S. Fiori, "Image compression using principal component neural networks," *Image and Vis. Computing*, vol. 19, no. 10, pp. 649-668, Aug. 2001.

[20] L. Ma and K. Khorsani, "Application of adaptive constructive neural networks to image compression," *IEEE Trans. on Neural Networks*, vol. 13, no. 5, pp. 1112-1126, Sept. 2002.

[21] R. Y. Li, J. Kim, and N. A. Shamakhi, "Image compression using transformed vector quantization," *Image and Vis. Computing*, vol. 20, no. 1, pp. 37-45, Jan. 2002.

[22] M. E. Peterson, D. Ridder, and H. Handels, "Image processing with neural networks-a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279-2301, Oct. 2002.

[23] T. Fang, J. Lu, Z. Wang, and Y. Sun, "Image compressing algorithm based on PCA/SOFM hybrid neural network," *IEEE Conf. on Ind. Electron. Soc.*, vol. 3, pp. 2103-2107, Nov. 2003.

[24] J. Robinson and V. Kecman, "Combining support vector machine learning wiyh discrete cosine transform in image compression," *IEEE Trans. on Neural Networks*, vol. 14, no. 4, pp. 950-958, Jul. 2003.

[25] Z. Wang, C. S. Leung, Y. S. Zhu, and T. T. Wong, "Data compression on the illumination adjustable images by PCA and ICA," Signal Process.: *Image Commun.*, vol. 19, no. 10, pp. 934-954, Nov. 2004.

[26] A. L. Chan, S. Z. Der, and N. M. Nasarbadi, "A joint compression-discrimination neural transformation applied to target detection," *IEEE Trans. on Syst. and Cybern.*, vol. 35, no. 4, pp. 670-681, Aug. 2005.

[27] A. M. Base, K. Jancke, A. Wismuller, S. Foo, and T. Martinetz, "Medical image compression using topology-preserving neural networks," *Eng. Applicat. of Artificial intell.*, vol. 18, no. 4, pp. 383-392, Jun. 2005.

[28] H. S. Soliman and M. Omari, "A neural networks approach to image data compression," *Appl. Soft Computing*, vol. 6, no. 3, pp. 258-271, Mar. 2006.

[29] I. Vilovic, "An experience in image compression using neural networks," *Int. Symp. on Multimedia Signal Process. and Commun.*, pp. 95-98, Jun. 2006.

[30] B. Northan and R. D. Dony, "Image compression with a multiresolution neural network," *Can. J. of Elect. and Comput. Eng.*, vol. 31, no. 1, pp. 49-58, Jun. 2006.

[31] S. A. Durai, and E. A. Saro, "Image compression with back-propagation neural network using cumulative distribution function," *World Academy of Sci., Eng. and Technol.*, vol. 17, pp. 60-64, 2006.

[32] P. Danchenko, F. Lifshits, I. Orion, S. Koren, and S. Mark, "NNIC-neural network image compressor for satellite positioning system," *Acta Astronautica*, vol. 60, no. 9, pp. 622-630, May 2007.

[33] B. Arunapriya and D. K. Devi, “Image compression using single layer linear neural networks,” *Int. Conf. and Exhibition on Biometrics Technol.*, vol. 2, pp. 345-352, 2010.

[34] S. Sahami and M. G. Shayesteh, “Bi-level image compression technique using neural networks,” *IET Image Process.*, vol. 6, no. 5, pp. 496-506, Jul. 2012.

[35] G. Boopathi and S. Arockiasamy, “Image compression: Wavelet transform using radial basis function (RBF) neural network,” *IEEE Annu. Indian Conf. (INDICON)*, pp. 340-344, Dec. 2012.

[36] M. Elarbi and C. B. Amar, “Image authentication algorithm with recovery capabilities based on neural networks in the DCT domain,” *IET Image Process.*, vol. 8, no. 11, pp. 619-626, Nov. 2013.

[37] S. Haykin, *Neural Networks and Learning Machines*, 3^{rd} ed. Upper Saddle River, NJ: Prentice Hall, 2009.

[38] B. Anjana and R. Shreeja, “Image compression an artificial neural network approach,” *Int. J. of Computational Eng. Research*, vol. 2, no. 8, pp. 53-58, Dec. 2012.

[39] X. Yao, “Evolving artificial neural networks,” *IEEE Proc.*, vol. 87, no. 9, pp. 1423-1447, Feb. 1999.

[40] J. Jiang, “Image compression with neural networks-A survey,” Signal Process: *Image Commun.*, vol.14, no. 9, pp. 737-760, Jul. 1999.

[41] V. C. Klema and A. J. Laub, “The singular value decomposition:its computation and some applications,” *IEEE Trans.on Autom. Control*, vol. 25, no. 2, pp. 164-176, Apr. 1980.

[42] J. Chen, T. Berger, and S. Hemamai, “Multiple description quantization via Gram Schmidt Orthogonalization,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 12, pp. 5197-5216, Dec. 2006.