Republic of Iraq
Ministry of Higher Education
and Scientific Research,
Al-Nahrain University,
College of Science, Department of Mathematics
and Computer Applications

Design and Implementation Image Compress and Decompress Wireless Network System

A thesis
Submitted to the Council of the College of Science / Al-Nahrain University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mathematics

By
Noor Salameh Shehdeh
(B.Sc., Math. / College of Science / Al-Nahrain University, 2014)

Supervised by
Prof. Dr. Ali Abdul Hafidh Ibrahim

October 2016 ***Dhul Hijjah 1437***

بسم الله الرحمن الرحيم

اللَّهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مَثَلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِنْ شَجَرَةٍ مُبَارَكَةٍ زَيْتُونَةٍ لَا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا يُضِيءُ وَلَوْ لَمْ تَمْسَسْهُ نَارٌ نُورٌ عَلَى نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَنْ يَشَاءُ وَيَضْرِبُ اللَّهُ الْأَمْثَالَ لِلنَّاسِ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ

صدق الله العظيم

سورة النور
الآيه (٣٥)

Dedication

To everyone who Lit up the mind of others by his knowledge…
Or guide the correct answer to the questioner who confusion of it…
And shown by his leniency the humility of scientists…
And by his roominess the leniency of the knowledgeable persons …

إلى من أضاء بعلمه عقل غيره...
أو هدى بالجواب الصحيح حيرة سائليه...
فأظهر بسماحته تواضع العلماء...
و برحابته سماحة العارفين...

Noor

Acknowledgements

Thanks to Allah for his help and for giving me the ability to complete this work.

I would like to express my sincere appreciation and gratitude to my supervisor Prof. Dr. Ali A. Ibrahim for his continuous help and advice during the preparation of this thesis.

My deep gratitude to the college of science and to the staff members of the department of mathematics and computer application of Al-Nahrain University especially Mrs. Iman A. Hussain to her support and help me.

Also, I would like to present the most beautiful words of thanks to my parents, my sister and my brother for their continuous help and encouragement.

Finally, I would like to thank my fellow students for the support they gave to me during my study.

Noor Salameh Shehdeh

September, 2016

Supervisor Certifications

I certify this thesis entitled (***Design and Implementation Image Compress and Decompress Wireless Network System***) was prepared by (***Noor Salameh Shehdeh***) under my supervision at the department of mathematics and computer applications, College of Science, Al-Nahrain University, in partial fulfillment of the requirements for the degree of Master of Science in Mathematics.

Signature:

Name: **Dr. Ali Abdul Hafidh Ibrahim**

Title: **Professor**

Date: **10 / 10 / 2016**

In view of the available recommendations, I forward this thesis for debate by the examining committee.

Signature:

Name: Dr. Osama H. Mohammed

Title: Assistant professor

Address: Head of Mathematics and Computer Application Department

Date: 10 / 10 / 2016

Examining Committee Certification

We certify that we have read this thesis entitled (***Design and Implementation Image Compress and Decompress Wireless Network System***) and we examined the student (***Noor Salameh Shehdeh***) in its contents and in what it connected with, and that, in our opinion, it meets the standards of a thesis for the degree of master of Science in Mathematics.

Signature:
Name: **Dr. Raid Kamel Naji**
Title: **Professor**
Date: **/ / 2017**
(Chairman)

Signature:
Name: **Dr. Jamil A. Ali Al-Hawasy**
Title: **Assistant professor**
Date: **/ / 2017**
(Member)

Signature:
Name: **Dr. Osama H. Mohammed**
Title: **Assistant professor**
Date: **/ / 2017**
(Member)

Signature:
Name: **Dr. Ali Abdul Hafidh Ibrahim**
Title: **Professor**
Date: **/ / 2017**
(Member and supervisor)

I, hereby certify upon the decision of the examining committee

Signature:
Name: **Dr. Hadi M. Abood**
Title: **professor**
Address: **Dean of the college of science**
Al-Nahrain University
Date: **/ / 2017**

Abstract

The goal of this thesis is to design and implementation image compress and decompress wireless network system. The proposal wireless network system consisting of one central Personal Computer (PC) and two Personal Computers (PCs) that communicate with each other through router device. The central PC takes the responsibility of monitoring and controlling the PCs of the network. All network PCs communicate with each other by Transmission Control Protocol / Internet Protocol (TCP/IP) protocol suit. In the central PC, the network administrator selected the required image and send it to the other PCs which one of it will compress the grayscale image and other will compress the color image using the following methods: Principle Component Analysis (PCA), Singular Value Decomposition (SVD), Hybrid (Discrete Cosine Transform (DCT) & Discrete Wavelet Transform (DWT)) and Backpropagation Neural Network (BPNN). A comparison between these image compression methods is made based on some of the well-known fidelity measurements such as Compression Ratio (CR) and Mean Square Error (MSE) which have been used to assess the quality of the reconstructed image also based on the computation time of running compression process. The hybrid (DCT & DWT) method yields better results since the resulted reconstructed image has a good quality because of a lower MSE and it gives a higher CR also it takes short time for running the compression process.

Table of Abbreviations

Abbreviation	Meaning
1-D	One Dimensional
2-D	Two Dimensional
AC	Approximation Coefficient
ANN	Artificial Neural Network
AP	Access Point
BP	Backpropagation
BPNN	Backpropagation Neural Network
CR	Compression Ratio
DC	Detail Coefficient
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FWT	Fast Wavelet Transform
IDCT	Inverse Discrete Cosine Transform
IDWT	Inverse Discrete Wavelet Transform
IFWT	Inverse Fast Wavelet Transform
IP	Internet Protocol
LAN	Local Area Network
MAN	Metropolitan Area Network
MSE	Mean Square Error
P2P	Point-to-Point
P2MP	Point-to-Multipoint
PC	Personal Computers
PCA	Principle Component Analysis
PSNR	Peak Single Noise Ratio
RGB	Red, Green, Blue
SV	Singular Value
SVD	Singular Value Decomposition
SQ	Scalar Quantization
TCP	Transmission Control Protocol
VQ	Vector Quantization
WAN	Wide Area Network
WDS	Wireless Distributed System

Table of Symbols

Symbol	Meaning
$\mathbf{X}$	p-variate random variable
$\boldsymbol{\mu}$	Mean
$\boldsymbol{\Sigma}$	Covariance matrix
$\sigma_{jj'}$ $j, j' = 1, 2, \dots, p$	Variance
α_k	The eigenvector corresponding to the k^{th} eigenvalue of Σ
$Y_{(k)}$	the k^{th}principal component
ρ	the correlation matrix
X_i , $i = 1, 2, \dots, n$	the i^{th} observation
σ_i $i = 1, \dots, n$	Singular value
λ	eigenvalue
u_j ,	eigenvectors of AA^T
v_j	eigenvectors of A^TA
$L^2(\mathbb{R})$	the space of all square integrable functions on $\mathbb{R}$
$\psi(x)$	Wavelet function
$\psi_{a,b}$	the dilated–translated wavelets
b	a translation parameter
a	a dilation or scale parameter
$a^{-1/2}$	a normalization constant
$\phi(x)$	scaling function
$W_f(a, b)$	The continuous wavelet transform of f
$h_\phi(n)$	the scaling function coefficients
h_ϕ	a scaling vector
$h_\psi(n)$	the wavelet function coefficients
h_ψ	The wavelet vector.
j_0	an arbitrary starting scale
y_k	The input into the j^{th} node
w_{kj}	Weights connections between layer k and j
θ	a threshold value
u_j	The output into the j^{th} node
x_j	the weighted sum of the inputs into the j^{th} node
α	the learning rate
η	the momentum
δ_j	error term
Δw_{kj}	the weight change for a weight connecting a node in layer k to a node in layer j

Table of Figures

Table of Diagrams

Table of Contents

Introduction

In recent years, the development and demand of multimedia product grows increasingly fast, contributing to insufficient bandwidth of network and storage of memory device. Therefore, Compression is useful because it helps reduce the consumption of expensive resources such as hard disk space or transmission bandwidth. Image compression is an application of data compression on digital images. Digital images contain large amount of digital information that need effective techniques for storing and transmitting large volume of data. Image compression techniques are used for reducing the amount of data required to represent a digital image [39] [27].

There are two types of image compression algorithm: Lossless and Lossy. In the lossless compression the compressed image is totally replica of the original input image, there is not any amount of loss present in the image. While in Lossy compression the compressed image is not same as the input image, there is some amount of loss is present in the image [29].

Numerous studies carried out on variety of mathematical methods have proven its advantages in image compression. In this work the following methods will be presented:

A. Principal Component Analysis (PCA) which is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a new set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables [12].

B. Singular Value Decomposition (SVD) is said to be a significant topic in linear algebra by many renowned mathematicians. SVD has many practical and theoretical values; Special feature of SVD is that it can be performed on any real $m \times n$ matrix A which refactoring a matrix A into three new matrices U ,Σ and V in such a way that U and V are orthogonal matrices, Σ is a diagonal matrix with singular value (SV) on the diagonal [38].

C. Discrete Cosine Transform (DCT) exploits cosine functions, it expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies and magnitudes. DCT is a technique for converting a signal into elementary frequency components. DCTs are important to numerous applications in science and engineering, from lossy compression of images (where small high-frequency components can be discarded), to spectral for the numerical solution of partial differential equations [9].

D. Discrete Wavelet Transform (DWT) is any wavelet transform for which the wavelets are discretely sampled, DWT decomposes a signal into a set of mutually orthogonal wavelet basis functions. These functions differ from sinusoidal basis functions in that they are spatially localized that is, nonzero over only part of the total signal length. Furthermore, wavelet functions are dilated, translated and scaled versions of a a common function ψ, known as the mother wavelet. DWT is invertible, so that the original signal can be completely recovered from its DWT representation [23].

Also, the Artificial Neural Network have proved to be useful in image compression. Artificial Neural Network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes [17].

Image communication through handheld mobile/portable multimedia devices over wireless networks (WI-FI and cellular networks) and Internet is increasing day by day. The use of wireless technology is rapidly becoming the most popular way to join a network. Wi-Fi technology is one of the several existing technologies that offer the convenience way of data transmitting without physical connection [31].

The aim of this work is to design and implement an image compress and decompress by using wireless network system. The main steps can be summarized as follows:

- Design and Implement compression methods to transfer images over wireless network.
- Make a comparison between the proposed image compression methods based on some of the well-known fidelity measures which have been used to assess the quality of the reconstructed image and execution time.
- Design and implement a wireless network system.
- Monitor and control the wireless network by using control tool software.

This thesis consists of three chapters.

In chapter one, a mathematical background to the methods that related to present work and artificial neural network will be introduced.

Chapter two introduces an overview of image compression and image compression methods that used in the present work also introduces an overview of wireless network.

Chapter three includes in details the designed and implemented the proposal system and the implementation results with general discussion.

Chapter One

Fundamental Information

Chapter One
Fundamental Information

1.1 Introduction:

This chapter consists of three sections. The first section includes a literature survey for techniques used for image compression, as well as, for that related to present work. In the second section, a mathematical background to the methods that relating to the proposed work in this thesis will be shown. In the third section the artificial neural network will be introduced in details.

1.2 Literature survey:

This work required research in two main fields, so the review is presented according to these fields, this review will focus on the researches in the last and recent years as shown in Diagram (1.1).

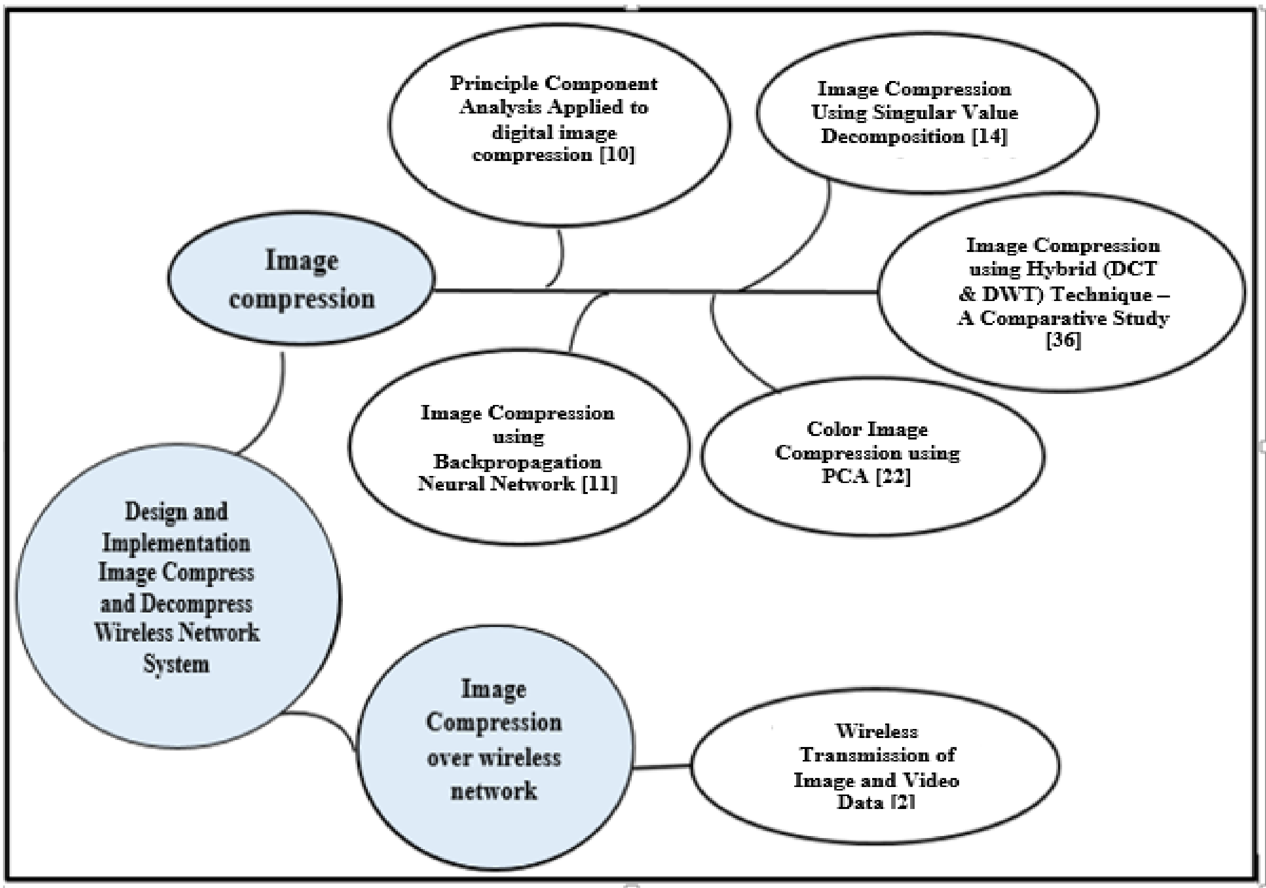

Diagram (1.1) image compression methods and image compression over Wireless network at 2002 and from 2012 to 2016.

In the field of image compression, Rafael in 2012 [5], described the use of a statistical tool Principal Component Analysis (PCA) for image compression, applying these concepts to digital images. The quantity of principal components used in the compression influences the recovery of the original image from the final (compacted) image.

Samruddhi and Reena in 2013 [14], presented one such image compression technique called Singular Value Decomposition (SVD). Basic mathematics of SVD is dealt with in detail and results of applying SVD on an image are also discussed. SVD is applied on variety of images for experimentation. The work is concentrated to reduce the number of eigenvalues required to reconstruct an image.

Nehain in 2014 [11], demonstrated the implementation of Backpropagation Neural Network (BPNN) algorithm on image compression system with good performance. The BPNN has been trained and tested for the analysis of different images.

Mohammad and Mostafa in 2015 [22], introduced a new method for compressing color images that utilizes the PCA method where it was introduced to compress a single image in RGB color space using the correlations between three Red, Green and Blue color domains

Ul Ain and Khaitan in 2016 [36], presented image compression using Hybrid Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) such that the advantages of DCT and DWT are used as proposed methodology. Several images are compared after compression by applying DCT, DWT and Hybrid (DCT & DWT), it was found that Hybrid gives the best results.

In image compression over wireless network field, Vladimir, at el. in 2002 [2], focused on the issues involved in wireless transmission of compressed image and video data.

1.3 Mathematical Background:

1.3.1 Principle Component Analysis, [19], [12]:

Principal Component Analysis (PCA) is a popular multivariate method in classical data analysis. A PCA computes uncorrelated linear combinations of the original variables so that the first linear combination has the largest variance, the second linear combination has the second largest variance, and so on. These linear combinations are called the principal components. Situations where PCA is most applicable include problems with high dimensional data and problems with highly correlated data. Suppose a data set has p variables where p is very large. Then, the principal components can be computed and the first $k(< p)$ principal components which contain most of the variation from the original variables are selected for further analysis. Then, analysis can be performed on these components without much loss of information. In this situation, PCA reduces the dimension of the data, hence reducing the computing power required to solve the problem. In another situation where correlation exists among the variables, PCA creates uncorrelated variables, hence eliminates the problem of collinearity. A PCA is typically used as an exploratory tool. After the principal components are computed, analysis and interpretation can then be performed on the principal components instead of the original variables.

Let $\boldsymbol{X} = \left(X_1 X_2, \dots, X_p\right)$ be a p-variate random variable from a distribution with mean $\boldsymbol{\mu} = \left(\mu_{1,}\mu_2, \dots, \mu_p\right)$ and variance-covariance $\boldsymbol{\Sigma} = \left[\sigma_{jj'}\right]$ for $j, j' = 1,2, \dots, p$. The first principal component is the linear combination Y_1 of $\boldsymbol{X}$ such that $Y_1 = \boldsymbol{\alpha}_1'\boldsymbol{X}$ has the largest variance under the constraint that $\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1 = 1$. The $\boldsymbol{\alpha}_1$ that maximizes $var(Y_1)$ under these conditions can be found using a Lagrange multiplier as follows. Since $var(Y_1) = var(\boldsymbol{\alpha}_1'\boldsymbol{X}) = \boldsymbol{\alpha}_1'\boldsymbol{\Sigma}\boldsymbol{\alpha}_1$, Let

$$\begin{aligned} \phi_1 &= var(Y_1) - \lambda(\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1 - 1) \\ &= \boldsymbol{\alpha}_1'\boldsymbol{\Sigma}\boldsymbol{\alpha}_1 - \lambda(\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1 - 1) \end{aligned}$$

Differentiate ϕ_1 with respect to $\boldsymbol{\alpha}_1$ and set the derivative equal to zero,

$$\frac{\partial \phi_1}{\partial \alpha_1} = 2(\boldsymbol{\Sigma}\boldsymbol{\alpha}_1 - \lambda\boldsymbol{\alpha}_1) = 0$$

Hence,

$$(\boldsymbol{\Sigma} - \lambda \boldsymbol{I})\boldsymbol{\alpha}_1 = 0 \tag{1.1}$$

Then, take the derivative of ϕ_1 with respect to λ and set this derivative to zero,

$$\frac{\partial \phi_1}{\partial \lambda} = (\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1 - 1)$$

This gives

$$\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1 = 1 \tag{1.2}$$

The λ and $\boldsymbol{\alpha}_1$ that satisfy both equations (1.1) and (1.2) are an eigenvalue-eigenvector pair of $\boldsymbol{\Sigma}$. Moreover, since

$$var(Y_1) = \boldsymbol{\alpha}_1'\boldsymbol{\Sigma}\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_1'\lambda\boldsymbol{\alpha}_1 = \lambda\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1 = \lambda$$

and Y_1 has the largest variance, then $\lambda = \lambda_1$ where λ_1 is the largest eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{\alpha}_1$ the eigenvector corresponding to λ_1. Therefore, the first principal component is a linear combination of $\boldsymbol{X}$ whose coefficients are the elements of the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{\Sigma}$.

The second principal component is a linear combination $Y_2 = \boldsymbol{\alpha}_2'\boldsymbol{X}$ with the constraint $\boldsymbol{\alpha}_2'\boldsymbol{\alpha}_2 = 1$. Moreover, Y_2 has the second largest variance and it is uncorrelated to Y_1, i.e., the covariance, $cov\,(Y_1, Y_2) = 0$. Again, a Lagrange multiplier is used to find $\boldsymbol{\alpha}_2$. Since $var(Y_2) = \boldsymbol{\alpha}_2'\boldsymbol{\Sigma}\boldsymbol{\alpha}_2$ and

$$cov(Y_1, Y_2) = \boldsymbol{\alpha}_1'\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 = \boldsymbol{\alpha}_2'\boldsymbol{\Sigma}\boldsymbol{\alpha}_1 = \lambda_1\boldsymbol{\alpha}_2'\boldsymbol{\alpha}_1 = \lambda_1\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_2 = 0, \tag{1.3}$$

$$\begin{aligned}\phi_2 &= var(Y_2) - \lambda(\boldsymbol{\alpha}_2'\boldsymbol{\alpha}_2 - 1) - \delta(\lambda_1\boldsymbol{\alpha}_2)'\boldsymbol{\alpha}_1 \\ &= \boldsymbol{\alpha}_2'\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - \lambda(\boldsymbol{\alpha}_2'\boldsymbol{\alpha}_2 - 1) - \delta(\lambda_1\boldsymbol{\alpha}_2)'\boldsymbol{\alpha}_1 \end{aligned} \tag{1.4}$$

Differentiate ϕ_2 of equation (1.4) with respect to $\boldsymbol{\alpha}_2$ and set it equal to zero,

$$\frac{\partial \phi_2}{\partial \alpha_2} = 2(\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - \lambda\boldsymbol{\alpha}_2) - \delta\lambda_1\boldsymbol{\alpha}_1 = 0 \tag{1.5}$$

Multiplying the left hand side of equation (1.5) by $\boldsymbol{\alpha}_1$ and then using equation (1.3), we obtain

$$2\boldsymbol{\alpha}_1'(\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - \lambda\boldsymbol{\alpha}_2) - \delta\lambda_1\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1 = \delta\lambda_1 = 0$$

Since $\lambda_1 \neq 0$, $\delta = 0$. Hence, equation (1.5) becomes

$$\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - \lambda\boldsymbol{\alpha}_2 = 0$$

Similar to the solution for the first principal component, $\boldsymbol{\alpha}_2$ is the eigenvector corresponding to the second largest eigenvalue, λ_2. That is, the second principal component, Y_2, is a linear combination of $\boldsymbol{X}$ whose coefficients are the elements of the eigenvector corresponding to the second largest eigenvalue of $\boldsymbol{\Sigma}$. The rest of the principal components are found in the same way. More formally, the k^{th} principal component is the linear combination

$$Y_k = \boldsymbol{\alpha}_k' \boldsymbol{X}$$

Where $\boldsymbol{\alpha}_k$ is the eigenvector corresponding to the k^{th} eigenvalue of $\boldsymbol{\Sigma}$ and cov $(Y_k, Y_{k'}) = 0$ for $k \neq k'$.

Vector $\boldsymbol{\alpha}_k$ is the vector of coefficients of principal component Y_k. The magnitude of element j of $\boldsymbol{\alpha}_k$, denoted by α_{jk}, indicates the significance of variable X_j to principal component Y_k. Coefficient α_{jk} is also proportional to the correlation between X_j and Y_k.

Let $\rho X_j, Y_k$ be the correlation between X_j and Y_k. Then,

$$\rho X_j, Y_k = \frac{\alpha_{jk}\sqrt{\lambda_k}}{\sqrt{\sigma_{jj}}} \qquad (1.6)$$

The correlation defined in equation (1.6) provides another avenue to understand the contribution of variable X_j to principal component Y_k. Unlike α_{jk} which indicates the importance of variable X_j to principal component Y_k when other variables are included, $\rho X_j, Y_k$ measures the importance of variable X_j to principal component Y_k individually.

In situations where it is appropriate to standardize the variance, principal components can be found using the correlation coefficient matrix $\boldsymbol{\rho} = [\rho_{jj'}]$ where

$$\rho_{jj'} = \frac{\sigma_{jj'}}{\sigma_j \sigma_{j'}}$$

for $j, j' = 1,2, \dots, p$ where $\sigma_{j=}\sqrt{\sigma_{jj}}$. In this case, $\lambda_1 > \lambda_2 > \cdots > \lambda_p$ are the eigenvalues and $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_p$ are their corresponding eigenvectors of the correlation matrix $\boldsymbol{\rho}$. The correlation measure between X_j and Y_k of equation (1.6) becomes

$$\rho X_j, Y_k = \alpha_{jk}\sqrt{\lambda_k}$$

because element ρ_{jj} of $\boldsymbol{\rho}$ is one for all $j = 1, \dots, p$.

When the population distribution is not known, estimation for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are used to compute the principal components. In the sample context, let $\boldsymbol{X}_i = (X_{i1}X_{i2}, \dots, X_{ip})$ be the i^{th} observation where $i = 1,2, \dots, n$ and p is the number of variables. Let $\boldsymbol{S}$ be the sample variance-covariance matrix of

$$\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_n)$$

Let

$$\boldsymbol{X}^* = (\boldsymbol{X}_1^*, \boldsymbol{X}_2^*, \dots, \boldsymbol{X}_n^*)$$

be the centered data matrix where

$$X_i^* = \left(X_{i1} - \bar{X}_1, X_{i2} - \bar{X}_2, \dots, X_{ip} - \bar{X}_p\right)'$$

and

$$\bar{X}_j = \frac{\sum_{i=1}^{n} X_{ij}}{n}$$

Then, $\boldsymbol{S}$ can be computed easily as,

$$\boldsymbol{S} = \frac{(\boldsymbol{X}^*)'(\boldsymbol{X}^*)}{n-1}$$

Let $\hat{\lambda}_1 > \hat{\lambda}_2 > \cdots > \hat{\lambda}_p$ be the eigenvalues of $\boldsymbol{S}$ and $\boldsymbol{v}_k$ be the eigenvector corresponding to $\hat{\lambda}_k$ for $k = 1, \dots, p$. Then, it turns out that the coefficients vector for the k^{th} principal component, Y_k, based on the data matrix $\boldsymbol{X}$ is $\boldsymbol{v}_k$.

The sampling distributions of $\hat{\lambda}_k$ and $\boldsymbol{v}_k$ are difficult to derive. The asymptotic results for it could be derived under the following assumptions: observations $\boldsymbol{X}_i$, $i = 1, \dots, n$, are a random sample from a normal distribution and the eigenvalues of the population covariance matrix $\boldsymbol{\Sigma}$ are distinct and positive. The following summary of these results:

1. Let $\boldsymbol{\Lambda}$ be the diagonal matrix of eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_p$ of; then, $\sqrt{n}(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda})$ is approximately $N_p(\boldsymbol{0}, 2\boldsymbol{\Lambda}^2)$.
2. Let

$$\boldsymbol{A}_k = \lambda_k \sum_{j=1, j\neq k}^{p} \frac{\lambda_j}{(\lambda_j - \lambda_k)^2} (\boldsymbol{\alpha}_j \boldsymbol{\alpha}_j');$$

then, $\sqrt{n}(v_k - \alpha_k)$ is approximately $N_p(0, A_k)$.

3. Each $\hat{\lambda}_k$ is distributed independently of the elements of the associated $\boldsymbol{v}_k$.

Based on these results, asymptotic estimation and inference for the eigenvalues and the eigenvectors of $\boldsymbol{\Sigma}$ can be performed.

1.3.2 Singular Value Decomposition, [18]:

In many applications, it is necessary either to determine the rank of a matrix or to determine whether the matrix is deficient in rank. Theoretically, we can use Gaussian elimination to reduce the matrix to row echelon form and then count the number of nonzero rows. However, this approach is not practical in finite-precision arithmetic. If A is rank deficient and U is the computed echelon form, then, because of rounding errors in the elimination process, it is unlikely that U will have the proper number of nonzero rows. In practice, the coefficient matrix A usually involves some error. This may be due to errors in the data or to the finite number system. Thus, it is generally more practical to ask whether A is close to a rank-deficient matrix. However, it may well turn out that A is close to being rank deficient and the computed row echelon form U is not.

Assume that A is an $m \times n$ matrix with $m \geq n$. (This assumption is made for convenience only; all the results will also hold if $m < n$). The singular value decomposition method determining how close A is to a matrix of smaller rank such that it factoring A into a product $U\Sigma V^T$, where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix, and Σ is $m \times n$ matrix whose off-diagonal entries are all zeros and whose diagonal elements satisfy:

$$\sigma_1 \geq \sigma_2 \geq \cdots . \geq \sigma_n \geq 0$$

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

The σ_i's determined by this factorization are unique and are called the *singular values* of A. The factorization $U\Sigma V^T$ is called the *singular value decomposition* of A. Next, the state of SVD theorem.

Theorem (1): (The SVD Theorem) [18]:

If matrix A is an $m \times n$ matrix, then A has a singular value decomposition.

Now that we have shown how a matrix A can be decompose into the product $U\Sigma V^T$, the following properties of the SVD will be considered:

1. The singular values $\sigma_1, \dots, \sigma_n$ of A are unique; however, the matrices U and V are not unique.
2. Since V diagonalizes AA^T, it follows that the $\boldsymbol{v}_j$'s are eigenvectors of $A^T A$.
3. Since $AA^T = U\Sigma\Sigma^T U^T$, it follows that U diagonalizes AA^T and that the $\boldsymbol{u}_j$'s are eigenvectors of AA^T.
4. Comparing the jth columns of each side of the equation
$$AV = U\Sigma$$
then get
$$A\boldsymbol{v}_j = \sigma_j \boldsymbol{u}_j \qquad j = 1, \dots, n$$
Similarly,
$$A^T U = V\Sigma^T$$
and hence
$$A^T \boldsymbol{u}_j = \sigma_j \boldsymbol{v}_j \text{ for } j = 1, \dots, n$$
$$A^T \boldsymbol{u}_j = \boldsymbol{0} \text{ for } j = n+1, \dots, m$$
The $\boldsymbol{v}_j$'s are called the right singular vectors of A, and the $\boldsymbol{u}_j$'s are called *the left singular vectors* of A.
5. If A has rank r, then
   (i) $\boldsymbol{v}_1, \dots, \boldsymbol{v}_r$ form an orthonormal basis for the range of A^T.
   (ii) $\boldsymbol{v}_{r+1}, \dots, \boldsymbol{v}_n$ form an orthonormal basis for the null space of A.
   (iii) $\boldsymbol{u}_1, \dots, \boldsymbol{u}_r$ form an orthonormal basis for the range of A.

(iv) $\boldsymbol{u}_{r+1}, \dots, \boldsymbol{u}_m$ form an orthonormal basis for the null space of A^T.

6. The rank of the matrix A is equal to the number of its nonzero singular values (where singular values are counted according to multiplicity).

7. In the case that A has rank $r < n$, if we set

$$U_1 = (\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_r), V_1 = (\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_r)$$

and define $\Sigma_1$as in (1.7), then

$$A = U_1 \Sigma_1 V_1^T \tag{1.7}$$

The factorization (1.7) is called *the compact form of the singular value decomposition* of A. This form is useful in many applications.

1.3.3 Discrete Cosine Transform, [16]:

The Discrete Cosine Transform (DCT) performs the cosine-series expansion. The purpose of DCT is to perform decorrelation of the input signal and to present the output in the frequency domain. The DCT is known for its high energy compaction property, meaning that the transformed signal can be easily analyzed using few low-frequency components. It turns out to be that the DCT is a reasonable balance of optimality of the input decorrelation and the computational complexity. This fact made it widely used in digital signal processing.

Formally, the DCT is an invertible function $F: \mathbb{R}^N \to \mathbb{R}^N$ or equivalently an invertible square $N \times N$ matrix. The formal definition for the DCT of one-dimensional (1-D) sequence of length N is given by the following formula:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{\pi(2x+1)u}{2N}\right), \ u = 0,1, \dots, N-1 \tag{1.8}$$

The inverse transformation is defined as:

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left(\frac{\pi(2x+1)u}{2N}\right), \ x = 0,1, \dots, N-1 \tag{1.9}$$

In both equations (1.8) and (1.9), $\alpha(u)$ is defined as:

$$\alpha(u) = \begin{Bmatrix} \frac{1}{\sqrt{N}} & if\ u = 0 \\ \sqrt{\frac{2}{N}} & if\ u \neq 0 \end{Bmatrix} \tag{1.10}$$

As can be seen from equation (1.8), the substitution of $u = 0$ yields $C(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x)$. Thus the first transform coefficient is the average value of the sample sequence which is the mean of the sample. This value is called the detail coefficient (DC) of the transform and all other transform coefficients are referred to as approximation coefficients (AC).

To fix ideas, ignore the $f(x)$ and $\alpha(u)$ component in equation (1.8). The plot of $\sum_{x=0}^{N-1} \cos\left(\frac{(2x+1)u\pi}{2N}\right)$ for $N = 8$ and varying values of u is shown in figure (1.2). In accordance with our previous observation, the first the waveform $(u = 0)$ renders a constant DC value, whereas, all other waveforms $(u = 1,2,\ldots,7)$ give waveforms at progressively increasing frequencies. These waveforms are called the *cosine basis function*. Note that these basis functions are orthogonal. Hence, multiplication of any waveform in Figure (1.1) with another waveform followed by a summation over all sample points yields a zero (scalar) value, whereas multiplication of any waveform in Figure (1.1) with itself followed by a summation yields a constant (scalar) value. Orthogonal waveforms are independent, that is, none of the basis functions can be represented as a combination of other basis functions.

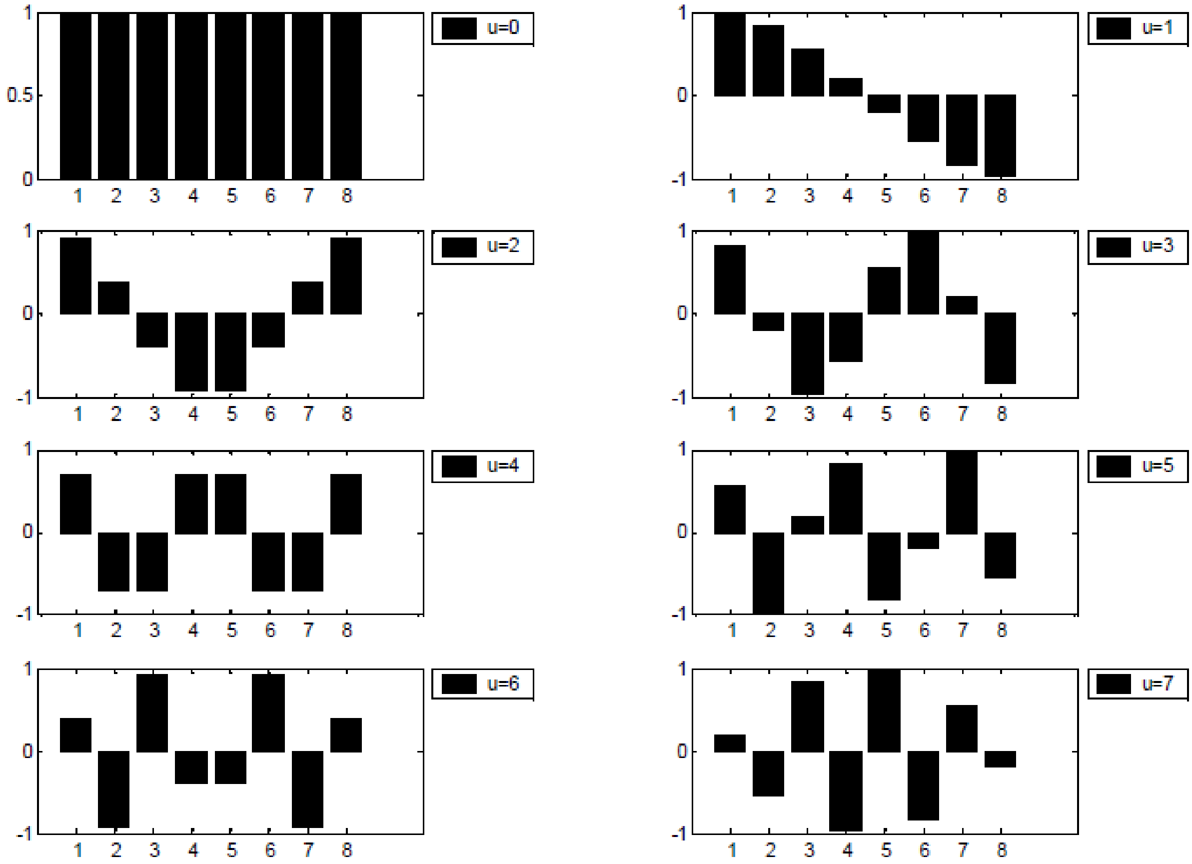

Figure (1.1) 1-D cosine basis functions for N = 8 [16].

If the input sequence has more than N sample points then it can be divided into sub-sequences of length N and DCT can be applied to these chunks independently. Here, a very important point to note is that in each such computation the values of the basis function points will not change. Only the values of $f(x)$ will change in each subsequence. This is a very important property, since it shows that the basis functions can be pre-computed offline and then multiplied with the sub-sequences. This reduces the number of mathematical operations (i.e., multiplications and additions) thereby rendering computation efficiency.

The two-dimensional (2-D) DCT is a direct extension of the 1-D case and is given by:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{\pi(2x+1)u}{2N}\right)\cos\left(\frac{\pi(2y+1)v}{2N}\right) \quad (1.11)$$

for $u, v = 0,1,2,\ldots,N-1$, $\alpha(u)$ and $\alpha(v)$ are defined in (1.10).

The inverse transformation defined as:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left(\frac{\pi(2x+1)u}{2N}\right)\cos\left(\frac{\pi(2y+1)v}{2N}\right) \qquad (1.12)$$

for $x, y = 0,1,2,\dots,N-1$.

The 2-D basis functions can be generated by multiplying the horizontally oriented 1-D basis functions (shown in Figure (1.1)) with vertically oriented set of the same functions. The basis functions for $N = 8$ are shown in Figure (1.2). It can be noted that the basis functions exhibit a progressive increase in frequency both in the vertical and horizontal direction. The top left basis function of results from multiplication of the DC component in Figure (1.1) with its transpose. Hence, this function assumes a constant value and is referred to as the DC coefficient.

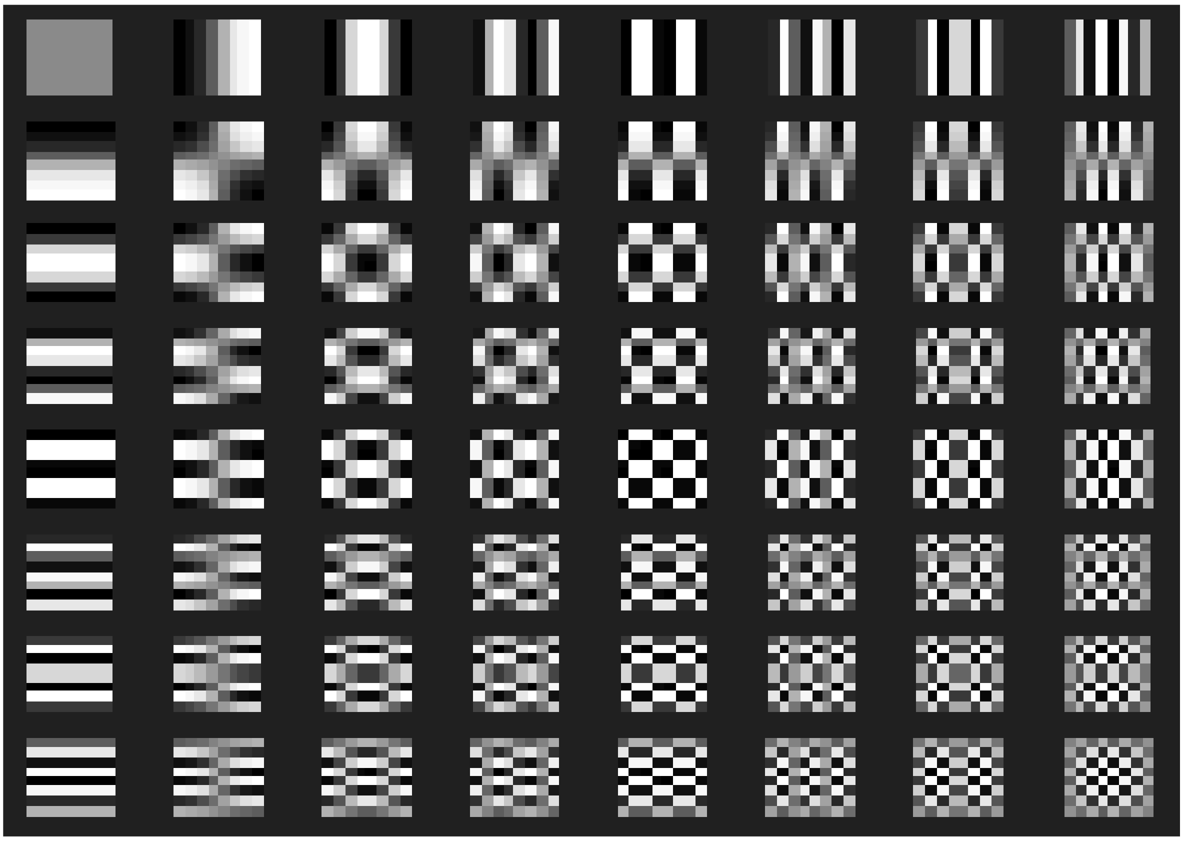

Figure (1.2) 2-D DCT basis functions for $N = 8$. Neutral gray represents zero, white represents positive amplitudes, and black represents negative amplitude [16].

1.3.4 Discrete Wavelet Transform, [1], [20], [23]:

Wavelet Transforms have become one of the most important and powerful tool of signal representation. It is based on small waves, called wavelets. Wavelets allow both time and frequency analysis of signals simultaneously because of the fact that the energy of wavelets is concentrated in time and still possesses the wave-like (periodic) characteristics. Wavelet transform is one of a best tool to determine where the low frequency area and high frequency area is.

Let $L^2(\mathbb{R})$ denote the space of all square integrable functions on $\mathbb{R}$. In signal processing parlance, it is the space of functions with finite energy. Let $\psi(x) \in L^2(\mathbb{R})$ be a fixed function. The function $\psi(x)$ is said to be wavelet if and only if its Fourier Transform $\Psi(\omega)$ satisfies:

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty \tag{1.13}$$

The equation (1.13) is called *admissibility condition*, which implies that the wavelet must have a zero average:

$$\int_{-\infty}^{\infty} \psi(x) dx = \Psi(0) = 0, \tag{1.14}$$

and therefore it must be oscillatory. In other words, ψ must be a sort of wave.

Let us now define the dilated–translated wavelets $\psi_{a,b}$ as the following functions:

$$\psi_{a,b} = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \tag{1.15}$$

Where $b \in \mathbb{R}$ is a translation parameter, whereas $a \in \mathbb{R}^+$ $(a \neq 0)$ is a dilation or scale parameter. The factor $a^{-1/2}$ is a normalization constant such that the energy, i.e., the value provided through the square inerrability of $\psi_{a,b}$, is the same for all scales a. One notices that the scale parameter a in (1.15) rules the dilations of the independent variable $(x - b)$. In the same way, the factor $a^{-1/2}$ rules the dilation in the values taken by ψ. With (1.15), one is able to decompose a square integrable function $f(x)$ in terms of these dilated–translated wavelets.

The continuous wavelet transform (CWT) of $f(x) \in L^2(\mathbb{R})$ is defined as:

$$W_f(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(x)\, \bar{\psi}_{a,b} dx = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(x) \bar{\psi}\left(\frac{x-b}{a}\right) dx, \qquad (1.16)$$

Where $\langle \, , \rangle$ is the scalar product in $L^2(\mathbb{R})$ defined as $\langle f, g \rangle = \int f(x) g(x) dx$. The CWT (1.16) measures the variation of f in a neighborhood of the point b, whose size is proportional to a. The inverse transform can be constructed by:

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^{\infty} W_f(a,b) \psi_{a,b}(x) \frac{da db}{a^2} \qquad (1.17)$$

Hence the CWT maps a one-dimensional function $f(x)$ to a function $W_f(a,b)$ of two continuous real variables a (*dilation*) and b (*translation*).

Since the input signal (e.g., a digital image) is processed by a digital computing machine, it is prudent to define the discrete version of the wavelet transform. Before we define the *discrete wavelet transform,* it is essential to define the wavelets in terms of discrete values of the *dilation* and *translation* parameters a and b instead of being continuous. There are many ways to discretize a and b and then represent the *discrete wavelets* accordingly. The most popular approach of discretizing a and b by using equation (1.18)

$$a = a_0^j, \qquad b = k b_0 a_0^j \qquad (1.18)$$

Where the parameter j and k are integers. Substituting a and b in equation (1.15) by equation (1.18), the discrete wavelets can be represented by:

$$\psi_{j,k}(x) = {a_0}^{-j/2} \psi\left({a_0}^{-j} x - k b_0\right) \qquad (1.19)$$

There are many choices to select the values of a_0 and b_0. We select the most common choice here: $a_0 = 2$ and $b_0 = 1$; hence, $a_0 = 2^j$ and $b = k2^j$. This corresponds to sampling (discretization) of a and b in such a way that the consecutive discrete values of a and b as well as the sampling intervals differ by a factor of two. This way of sampling is popularly known as *dyadic sampling* and the corresponding decomposition of the signals is called the *dyadic decomposition.* Using these values, we can represent the discrete wavelets as in equation (1.20), which constitutes a family of orthonormal basis functions,

$$\psi_{j,k}(x) = 2^{-j/2} \psi\left(2^{-j} x - k\right) \qquad (1.20)$$

and we can also write the scaling function as:

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k) \quad (1.21)$$

Here, k determines the position of $\psi_{j,k}(x)$ and $\phi_{j,k}(x)$ along the x-axis; and $2^{j/2}$ controls their height or amplitude. By choosing the scaling function $\phi(x)$ wisely, $\{\phi_{j,k}(x)\}$ can be made to span $L^2(\mathbb{R})$.

Generally, we will denote the subspace spanned over k for any j as:

$$V_j = \overline{\underset{k}{Span}\{\phi_{j,k}(x)\}}$$

We can verify that the size of V_j can be increased by increasing j, allowing functions with finer detail. There are four fundamental requirements of multiresolution analysis that scaling function and wavelet function must follow:

1. The scaling function is orthogonal to its integer translates.
2. The subspaces spanned by the scaling function at low resolutions are contained within those spanned at higher resolutions:

   $$V_{-\infty} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_{+\infty}$$

3. The only function that is common to all V_j is $f(x) = 0$. That is

   $$V_{-\infty} = \{0\}$$

4. Any function can be represented with arbitrary precision. As the level of the expansion function approaches infinity, the expansion function space V contains all the subspaces.

   $$V_{+\infty} = \{L^2(\mathbb{R})\}$$

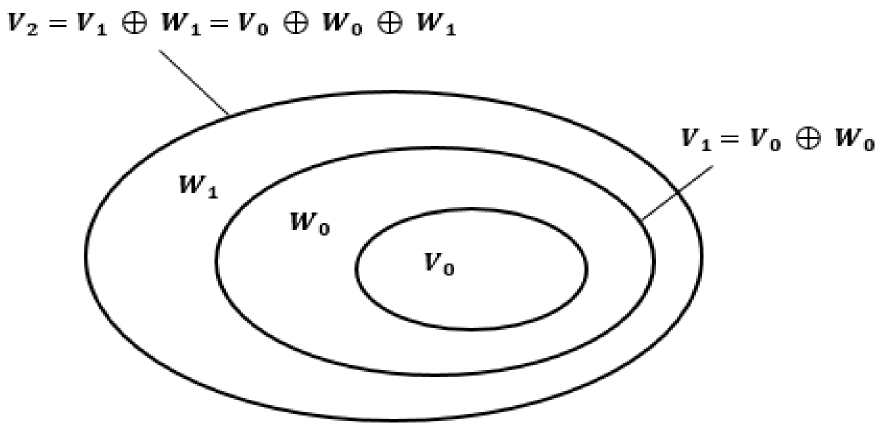

Figure (1.3) The relationship between scaling and wavelet function space [20].

Under these conditions, the expansion functions of subspace V_j can be expressed as a weighted sum of the expansion functions of subspace V_{j+1}

$$\phi_{j,k}(x) = \sum_n \alpha_n \phi_{j+1,k}(x) \tag{1.22}$$

Substituting for $\phi_{j+1,k}(x)$ and changing variable α_n to $h_\phi(n)$, this becomes

$$\phi(x) = h_\phi(n)\sqrt{2}\phi(2x-k) \tag{1.23}$$

where the $h_\phi(n)$ are called the scaling function coefficients and h_ϕ is referred to as a scaling vector. On the other hand, for all $k \in \boldsymbol{Z}$ we denote the subspace spanned by discrete wavelet set as:

$$W_j = \overline{\underset{k}{Span}\{\psi_{j,k}(x)\}}$$

And we define the discrete wavelet set $\psi_{j,k}(x)$ spans the difference between any two adjacent scaling subspaces, V_j and V_{j+1}. It is shown in Figure (1.3) and related by:

$$V_{j+1} = V_j \oplus W_j$$

where $\oplus$ denotes the union of spaces. We can thus express the space of $L^2(\mathbb{R})$ as:

$$L^2(\mathbb{R}) = V_{j_0} \oplus W_0 \oplus W_1 \oplus \ldots$$

This equation can even be extended as following:

$$L^2(\mathbb{R}) = \cdots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \ldots$$

which eliminates the scaling function and represents a function in terms of wavelets alone. If a function $f(x)$ is an element of V_1, an expansion contains an approximation of $f(x)$ using V_0 and the wavelets from W_0 encode the difference between this approximation and the actual function. Generally, we start from an arbitrary scale j_0. Then the equation can be written as:

$$L^2(\mathbb{R}) = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \ldots$$

Moreover, any wavelet function can be expressed as a weighted sum of shifted, double-resolution scaling functions. That is, we can write

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\phi(2x-n) \tag{1.24}$$

where the $h_\psi(n)$ are called the *wavelet function coefficients* and h_ψ is the *wavelet vector*. If the function being expanded is a sequence of numbers, like samples of a continuous function $f(x)$, the resulting coefficients are called the DWT of $f(x)$. By applying the principle of series expansion, the DWT coefficients of $f(x)$ are defined as:

$$W_{\phi}(j_0, k) = \frac{1}{\sqrt{M}} \sum_x f(x)\, \phi_{j_0,k}(x) \tag{1.25}$$

$$W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_x f(x)\, \psi_{j,k}(x) \tag{1.26}$$

for $j \geq j_0$ and the parameter M is a power of 2 which range from 0 to $j - 1$.

The function $f(x)$ can now expressed as:

$$f(x) = \frac{1}{\sqrt{M}} \sum_k W_{\phi}(j_0, k)\phi_{j_0,k}(x) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_{\psi}(j, k)\psi_{j,k}(x) \tag{1.27}$$

Where $\frac{1}{\sqrt{M}}$ acts as a normalizing factor.

The 2-D wavelet transforms are slightly different to 1-D ones. One can easily extend it by simply multiply the 1-D scaling and wavelet functions. Wavelet transform in 2-D is used in image processing. For 2-D wavelet transform, we need one 2-D scaling function, $\phi(x, y)$, and three 2-D wavelet functions, $\psi^H(x, y)$, $\psi^V(x, y)$ and $\psi^D(x, y)$. Each is the product of a 1-D scaling function ϕ and corresponding wavelet ψ. These are shown as follows:

$$\phi(x, y) = \phi(x)\phi(y) \tag{1.28}$$

$$\psi^H(x, y) = \psi(x)\phi(y) \tag{1.29}$$

$$\psi^V(x, y) = \phi(x)\psi(y) \tag{1.30}$$

$$\psi^D(x, y) = \psi(x)\psi(y) \tag{1.31}$$

For image processing, these functions measure the variation of intensity for the image along different directions: ψ^H measures variations along columns, ψ^V measures variations along rows, and ψ^D measures variations along diagonals. The scaling function ϕ gives the approximation as same as the 1-D one. When the scaling function and wavelet functions are given, extension of the 1-D DWT to 2-D is straightforward. We first define the basis functions:

$$\phi_{i,m,n}(x, y) = 2^{j/2}\phi\left(2^j x - m, 2^j y - n\right) \tag{1.32}$$

$$\psi^i_{j,m,n}(x, y) = 2^{j/2}\psi^i\left(2^j x - m, 2^j y - n\right), i \in \{H, V, D\} \tag{1.33}$$

Where the index i defines the direction of the wavelet functions. The 2-D DWT of function $f(x,y)$ of size $M \times N$ is defined as follows:

$$W_{\phi}(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, \phi_{j_0,m,n}(x,y) \tag{1.34}$$

$$W_{\psi}^{i}(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \psi_{j,m,n}^{i}(x,y), \quad i = \{H, V, D\} \tag{1.35}$$

Similarly, the variable j_0 is an arbitrary starting scale and $W_{\phi}(j_0, m, n)$ define the approximation of $f(x,y)$. Moreover, $f(x,y)$ can be obtained by 2-D inverse discrete wavelet transform (IDWT) defined by:

$$f(x,y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_{\phi}(j_0, m, n) \phi_{j_0,m,n}(x,y) +$$

$$\frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_{\psi}^{i}(j, m, n) \psi_{j,m,n}^{i}(x,y) \tag{1.36}$$

The efficient implementation of the DWT is the *fast wavelet transform* (FWT). By finding the relationship coefficients of the DWT at adjacent scales, we can reduce the compute complexity. Consider the multiresolution refinement equation:

$$\phi(x) = \sum_n h_{\phi}(n) \sqrt{2} \phi(2x - n) \tag{1.37}$$

By a scaling of x by 2^j, translation of x by k units, and letting $m = 2k + n$, we would get:

$$\begin{aligned} \phi\left(2^j x - k\right) &= \sum_n h_{\phi}(n) \sqrt{2} \phi\left(2\left(2^j x - k\right) - n\right) \\ &= \sum_m h_{\phi}(m - 2k) \sqrt{2} \phi\left(2^{j+1} x - m\right) \end{aligned} \tag{1.38}$$

and similarly

$$\psi\left(2^j x - k\right) = \sum_m h_{\psi}(m - 2k) \sqrt{2} \psi\left(2^{j+1} x - m\right) \tag{1.39}$$

Now consider the DWT coefficient functions $W_{\psi}(j, k)$. By changing variable we can get:

$$W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_x f(x) 2^{j/2} \psi\left(2^j x - k\right) \tag{1.40}$$

which, upon replacing $\psi\left(2^j x - k\right)$, it becomes

$$W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_x f(x) 2^{j/2} \left[\sum_m h_{\psi}(m - 2k) \sqrt{2} \psi\left(2^{j+1} x - m\right)\right] \tag{1.41}$$

rearranging the terms then gives

$$W_{\psi}(j,k)=\sum_m h_{\psi}(m-2k)\left[\frac{1}{\sqrt{M}}\sum_x f(x)2^{(j+1)/2}\sqrt{2}\psi\left(2^{j+1}x-m\right)\right] \qquad (1.42)$$

where the bracketed quantity is identical to $W_{\phi}(j_0,k)$ with $j_0=j+1$. We can thus write

$$W_{\psi}(j,k)=\sum_m h_{\psi}(m-2k)W_{\phi}(j+1,k) \qquad (1.43)$$

similarly, the approximation coefficients is written by

$$W_{\phi}(j,k)=\sum_m h_{\phi}(m-2k)W_{\phi}(j+1,k) \qquad (1.44)$$

We can find that both $W_{\psi}(j,k)$and $W_{\phi}(j,k)$ can be obtained by convolving $W_{\phi}(j+1,k)$, the scale $j+1$ approximation coefficients, with time-reversed scaling and wavelet vectors $h_{\psi}(-n)$ and $h_{\phi}(n)$, and downsampling the results by 2. It is a surprising and useful relationship. Diagram (1.2) illustrates the construction of FWT using filter bank.

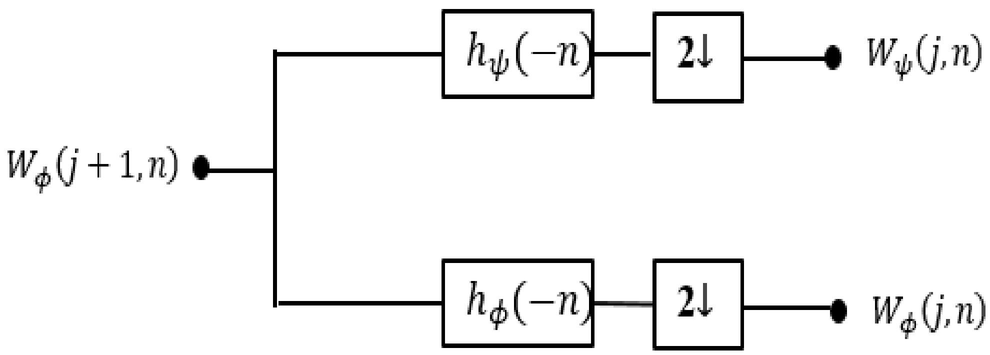

Diagram (1.2) The FWT analysis filter bank [20].

Reconstruction of $f(x)$ can also be formulated by using the scaling and wavelet vectors employed in the forward transform. It is called the inverse fast wavelet transform (IFWT). To generate the level $j+1$ approximation coefficients, it is a little bit different to FWT. We first upsample the approximation and detail coefficients of level j and then summing the result which are passed through the scaling and wavelet vector $h_{\psi}(n)$ and $h_{\phi}(n)$. This system is then depicted in Diagram (1.3).

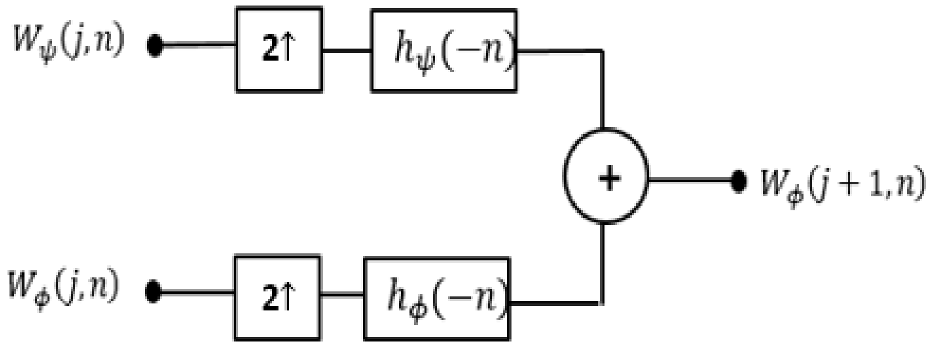

Diagram (1.3) The IFWT synthesis filter bank [20].

The 2-D wavelet transform also has its fast algorithm which is similar to the 1-D one. With separable 2-D scaling and wavelet functions, we simply take the 1-D FWT of the rows of $f(x, y)$, followed by the 1-D FWT of the resulting columns as shown in Diagram (1.4). Like the 1-D FWT, the 2-D FWT is constructed of approximation and detail part. As we can see in Diagram (1.5), the original image (matrix) can be decomposed into four subimages, which are W_ϕ , W_ψ^H , W_ψ^V and W_ψ^D. We can again divide the scale $j+1$ ACs into four parts in smaller size. In other words, the $j+1$ ACs are constructed by the scale j approximation and detail coefficients. The reconstruction algorithm is similar to the 1-D one. At each iteration, four scale j approximation and detail coefficients are upsampled and convolved with two 1-D filters, one is for the subimages' rows and the other is for its columns. Adding the results then we can obtain the $j+1$ approximation coefficients. By repeating the process, we can ultimately reconstruct the original image (matrix). Diagram (1.6) shows the synthesis filter bank of this operation.

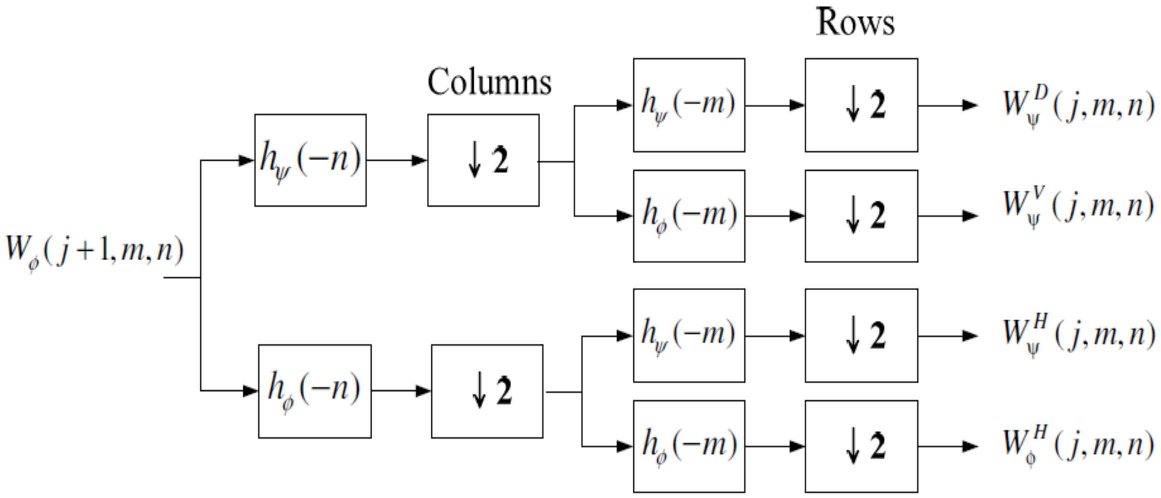

Diagram (1.4) The 2-D FWT analysis filter bank [20].

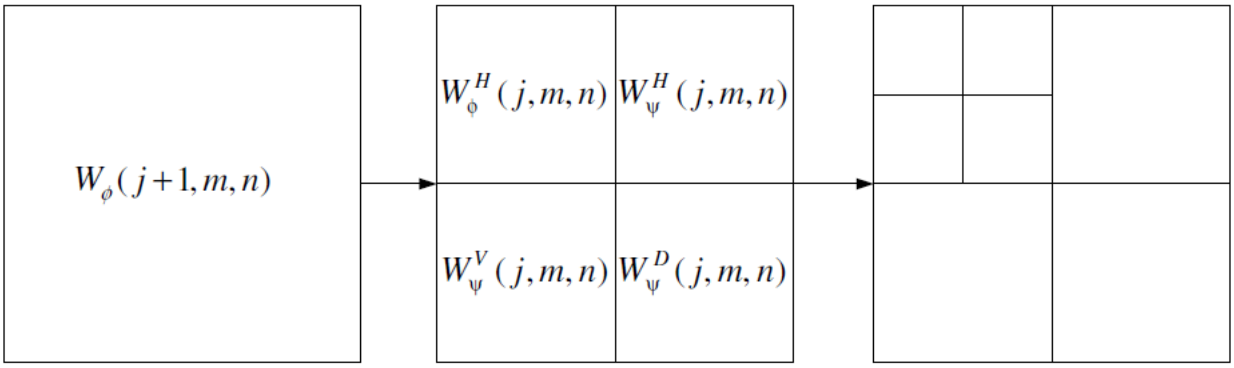

Diagram (1.5) The resulting decomposition [20].

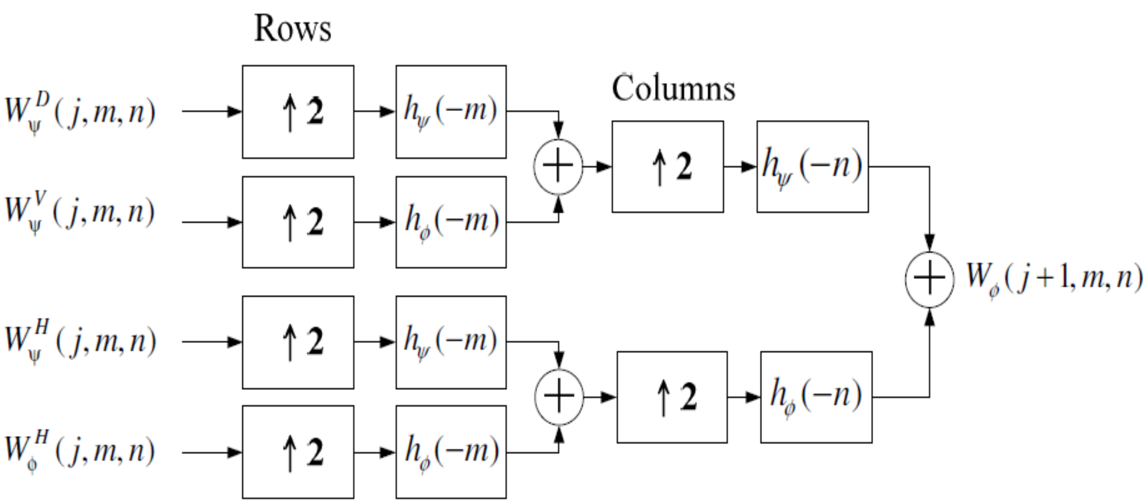

Diagram (1.6) The 2-D IFWT synthesis filter bank [20].

1.4 Artificial Neural Network:

An Artificial Neural Network (ANN) is an information-processing theory that is inspired which is the way of biological nervous systems, just like as the human brain process information. The key element of this concept is the fresh structure of the information processing system [22].

1.4.1 Artificial Neuron, [17], [34]:

Neural Networks are modeled after biological neural network structures. The neuron model shown in Figure (1.4) is the one that widely used in ANNs with some minor modifications on it.

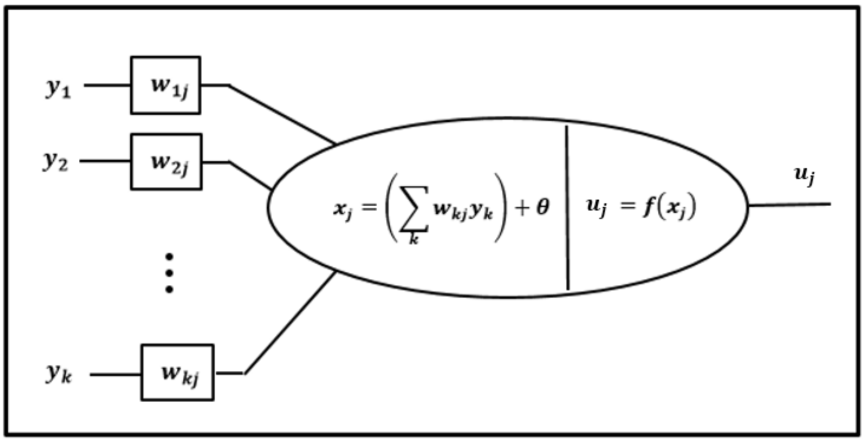

Figure (1.4) Artificial Neuron Structure [17].

The model artificial neuron shown here consistes of several inputs $y_1, y_2, \dots, y_k$. Each line connecting these inputs to the neuron is assigned a *weight*, which are denoted as $w_{1j}, w_{2j}, \dots, w_{kj}$ respectively. Weights in the artificial model correspond to the synaptic connections in biological neurons. The *threshold* in artificial neuron is usually represented by θ and the *activation* corresponding to the graded potential is given by the formula:

$$x_j = \sum_k w_{kj}\, y_k + \theta \qquad (1.45)$$

The inputs and the weights are real values. A negative value for a weight indicates an inhibitory connection while a positive value indicates an excitatory one. Although in biological neurons, has a negative value, it may be assigned a positive value in artificial neuron models. Sometimes, the threshold is combined for simplicity into the summation part by assuming an imaginary input $y_0 = +1$ and a connection weight $w_{0j} = \theta$. Hence the activation formula becomes:

$$x_j = \sum_k w_{kj}\, y_k \tag{1.46}$$

The output value of the neuron is a function of its activation in an analogy to the firing frequency of the biological neurons:

$$u_j = f\left(x_j\right) \tag{1.47}$$

There are a number of functions that are used. Some include binary threshold, linear threshold, sigmoid, hyperbolic tan and Gaussian.

1.4.2 Network Architectures [10]:

There are a number of architectures in use for ANNs. In feedforward neural networks, the neurons are organized in the form of layers. The neurons in a layer get input from the previous layer and feed their output to the next layer. In this kind of networks connections to the neurons in the same or previous layers are not permitted. The last layer of neurons is called the output layer and the layers between the input and output layers are called the hidden layers. The input layer is made up of special input neurons, transmitting only the applied external input to their outputs. In a network if there is only the layer of input nodes and a single layer of neurons constituting the output layer then they are called single layer network as shown in Figure (1.5).

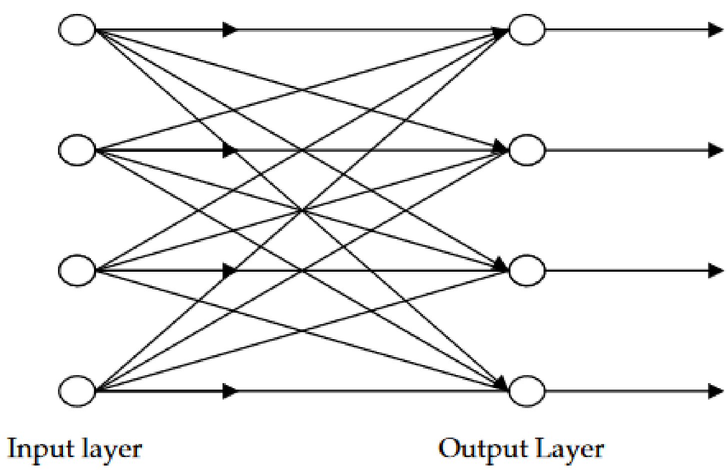

Figure (1.5) A Single layer feedforward network.

If there are one or more hidden layers, such networks are called multilayer networks as shown in Figure (1.6).

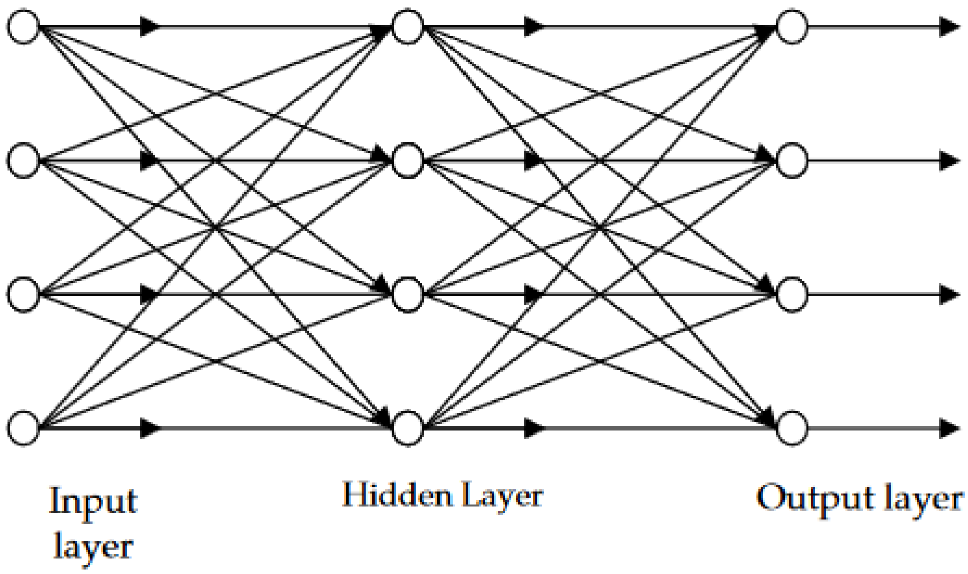

Figure (1.6) A multilayer feed forward network.

The structures, in which connections to the neurons of the same layer or to the previous layers are allowed, are called recurrent networks as shown in Figure (1.7).

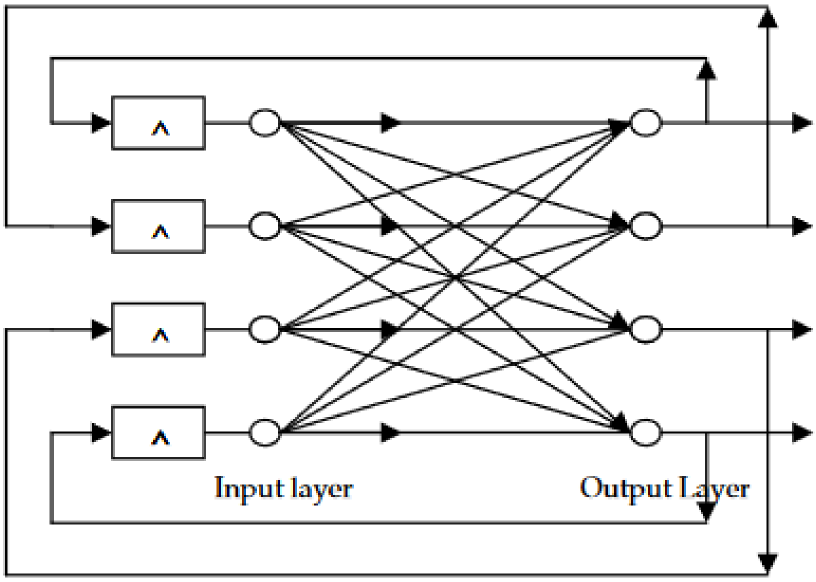

Figure (1.7) A recurrent network.

1.4.3 Learning in Artificial Neural Networks [10], [34]:

The ability to learn is a fundamental trait of intelligence. The process of learning in ANN can be thought of as the problem of amending the network model and the interconnecting weights, so that it can perform a given task effectively. There are many algorithms developed by researchers to train the neural network. These algorithms can be classified as one of three basics types:

i. **Supervised learning** or learning with a teacher, the network is provided with a correct answer (output) for every input pattern. Weights are determine to allow the network to produce answers as close as possible to the known correct answers.

ii. **Unsupervised learning** or learning without teacher, does not require a correct answer associated with each input pattern in the training data set. It explores the underlying structure in the data, and organizes patterns into categories from these correlations.

iii. **Hybrid learning** combines Supervised and Unsupervised learning. Parts of weights are usually determined through supervised learning, while the others are obtained through unsupervised learning.

1.4.4 Backpropagation Neural Network Algorithm, [21], [24]:

The backpropagation (BP) learning method can be applied to any multilayer network that uses differentiable activation functions and supervised training. It is an optimization procedure based on gradient descent that adjusts weights to reduce cost functions.

In order to derive BP algorithm, multilayer feedforward network having single hidden layer considered as shown in Figure (1.8).

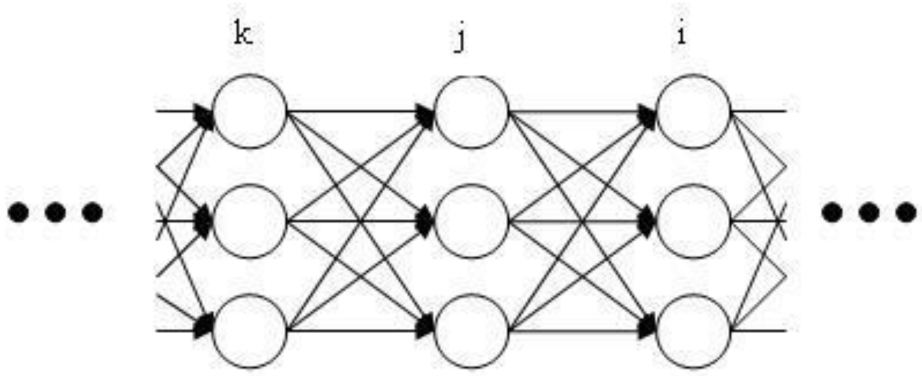

Figure (1.8) Multilayer feed forward Backpropagation Network.

The system has an input layer that is fully connected to the hidden layer and each unit in the hidden layer is fully connected to the output layer units. Weight connections between input layer unit k and hidden layer unit j are denoted by w_{kj}. While weight connections between hidden layer unit j and output unit i are denoted as w_{ji}. In the derivation of the backpropagation algorithm below the sigmoid function $f(x_j) = \frac{1}{1+e^{-x_j}}$, was used largely.

In preparation for the derivation of the algorithm, we need to define a measure of error. Intuitively, the error is the difference between the actual activation of an output node and the desired ("target") activation (t_j) for that node. The total error is the sum of these errors for each output node. Furthermore, since we'd like negative and positive errors not to cancel each other out, we square these differences before summing. Finally, we scale this quantity by a factor of $1/2$ for convenience (which will become clear shortly):

$$E = \frac{1}{2}\sum_j \left(t_j - u_j\right)^2 \tag{1.48}$$

It must be stressed that this equation applies only when the j^{th} layer is the output layer, which is the only layer for which the error is defined.

From the previous considerations mooted, the weight change for a weight connecting a node in layer k to a node in layer j is:

$$\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}} \tag{1.49}$$

Here α is a free parameter (the "learning rate") that we set prior to training; it lets us scale our step size according to the problem at hand. Note the negative sign: this indicates that weight changes are in the direction of decrease in error. Now expanding the partial derivative by the chain rule, we find:

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial u_j}\frac{\partial u_j}{\partial x_j}\frac{\partial x_j}{\partial w_{kj}} \tag{1.50}$$

where we recall from equation (1.46) that x_j is the weighted sum of the inputs into the j^{th} node, and hence

$$\frac{\partial x_j}{\partial w_{kj}} = y_k \tag{1.51}$$

We will find it notationally convenient to treat the first two factors as a single quantity, an "error term":

$$\delta_j = -\frac{\partial E}{\partial u_j}\frac{\partial u_j}{\partial x_j} \tag{1.52}$$

Nevertheless, we would like to calculate this term. We recall that $u_j = f\left(x_j\right)$ is the sigmoid function, the derivative of which is calculated in the following equation:

$$\begin{aligned}\frac{\partial u_j}{\partial x_j} = \frac{\partial f(x_j)}{\partial x_j} &= \frac{0.\left(1-e^{-x_j}\right)-\left(-e^{-x_j}\right)}{\left(1-e^{-x_j}\right)^2} \\ &= \frac{1}{1+e^{-x_j}}\left(\frac{e^{-x_j}}{1+e^{-x_j}}\right) \\ &= \frac{1}{1+e^{-x_j}}\left(1-\frac{1}{1+e^{-x_j}}\right) \\ &= u_j(1-u_j)\end{aligned} \tag{1.53}$$

Finally we consider the first partial derivative in the error term. When j is an output layer, this quantity is easy to calculate; it is just the derivative of equation (1.48) with respect to u_j, i.e.,

$$\frac{\partial E}{\partial u_j} = -(t_j - u_j) \tag{1.54}$$

(Notice how the scaling factor was neatly eliminated in taking the derivative.) Putting equations. (1.51), (1.53), and (1.54) into equation (1.50), we have:

$$\frac{\partial E}{\partial w_{kj}} = -(t_j - u_j)u_j(1-u_j)y_k \tag{1.55}$$

In the case where j is a hidden layer, $\partial E/\partial u_j$ is not quite as simple. Intuitively, we need to see how the error caused by u_j has propagated into the activations of the next (i^{th}) layer. Mathematically, this amounts to another application of the chain rule. Recalling our multi-variable calculus:

$$\frac{\partial E}{\partial u_j} = \sum_i \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial x_i}\frac{\partial x_i}{\partial u_j} \quad (1.56)$$

The first two partial derivatives are, from equation (1.52), just the error term for the next (i^{th}) layer, δ_i. The final term is simply the derivative of equation (1.46) with respect to the input:

$$\frac{\partial x_i}{\partial u_j} = w_{ji} \quad (1.57)$$

Making these substitutions yields:

$$\frac{\partial E}{\partial u_j} = -\sum_i \delta_i\, w_{ji} \quad (1.58)$$

so that in the case of hidden layers,

$$\frac{\partial E}{\partial w_{kj}} = -\sum_i (\delta_j\, w_{ji}) u_j (1 - u_j) y_k \quad (1.59)$$

Notice that even though this formula differs from the output-layer case, equation (1.55), we can write a single formula by employing our definition of the error term, equation (1.52):

$$\frac{\partial E}{\partial w_{kj}} = -\delta_j y_k \quad (1.60)$$

We've just buried the differences in the equations for the output and hidden layers in the error term. This is essentially what we will do in programming assignment: create a single function that calculates the weight change for the weight connecting a particular pair of nodes, given the activation of the upstream node and the error term of the downstream node. Each node will calculate its own error term, and this will vary according to whether the node lives in a hidden or output layer. And put all these pieces together.

From the above derivation, BP algorithm can be summarized as follows:

i. Initialize the weights to small random values.

ii. Select a training vector pair (input and the corresponding output) from the training set and present the input vector to the inputs of the network.

iii. Calculate the actual outputs this is the forward phase.

iv. According to the difference between actual and desired outputs (error). Adjust the weights of output and hidden layer to reduce the difference this is the backward phase.

v. Repeat from step ii for all training vectors.

vi. Repeat from step ii until the error is acceptably small.

The formulae for a weight connecting a node in layer k to a node in layer j as shown in Figure (1.12), the change in weight is given by:

$$\Delta w_{kj}(n) = \alpha \delta_j y_k + \eta \Delta w_{kj}(n-1) \tag{1.61}$$

Where:

- α is the learning rate, a real value on the interval (0,1];
- y_k is the activation of the node in layer k, i.e. the activation of the presynaptic node, the one upstream of the weight;
- n and $n-1$ refer to the iteration through the loop (i.e., the "epoch");
- η is the momentum, a real value on the interval [0,1); and
- δ_j is the "error term" associated with the node after the weight, i.e. the postsynaptic node.

Now, if j is the output layer,

$$\delta_j = (t_j - u_j)u_j(1-u_j) \tag{1.62}$$

if j is a hidden layer, then

$$\delta_j = \left(\sum_i \delta_i w_{ji}\right)u_j(1-u_j) \tag{1.63}$$

One can see from equation (1.63) that calculation of the error term for a node in a hidden layer requires the error term from nodes in the subsequent layer, and so on until the output layer error terms are calculated using (1.62). Thus, computation of the error terms must proceed backwards through the network, beginning with the output layer and terminating with the first hidden layer. It is this backwards propagation of error terms after which the algorithm is named.

Chapter Two

Theoretical Background

Chapter Two
Theoretical Background

2.1 Introduction:

This chapter consists of three sections. In the first section, an introduction to image and its definition in mathematics will be introduced. The second section, presented an overview on image compression and its algorithms, the basic image compression system in details and the various performance measurement parameters for comparing the compression techniques. The third section includes an overview of wireless network system.

2.2 Image [8]:

An images may be defined as a two dimensional function $f(x,y)$, where x and y are spatial (plane) coordinates and the amplitude of f at any pair of coordinates is called the intensity of the image at that point. If the size of image was indicated by $M \times N$, this means that the image array contains M rows and N columns, respectively. Such as an image can be represented by the matrix form given below:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ & & \vdots & \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

Each element of this array is called an *image element*, *picture element*, *pixel*, or *pel*.

2.3 Image Compression:

Images play an important role in the world of multimedia and its transmission with storage has become really a big burden as it occupy more space in memory. The goal of image compression is to create smaller files that use less space to store and less time to send. Image compression involves reducing the size of image data files, while retaining necessary information [32].

2.3.1. Image Compression Algorithms:

Generally, image compression algorithms are divided into two categories; i.e. lossless and lossy algorithms, as will describe below.

The Lossless compression allows exact reconstruction of the original image from its compressed form. In lossless compression, the compressed data of an image can be exactly restored. In this algorithm, the reconstructed image is identical to that original image. Lossless compression algorithm provide full reconstruction of the original data without any distortion in the image. Lossless image compression is widely used in many applications such as geophysics, telemetry, nondestructive evaluation, and medical imaging where no information loss is allowed during compression [28].

In Lossy compression a perfect reconstruction of the image is sacrificed by the elimination of some amount of redundancies in the image to achieve higher compression ratio. However, no visible loss of information is perceived under normal viewing conditions. In this compression, the original image cannot be exactly reconstructed from the compressed image. Lossy compression algorithm is widely used in many applications such as broadcast TV, video, and facsimile transmission [3].

2.3.2 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. Now consider an encoder and a decoder as shown in Figure (2.1). When the encoder receives the original image file, the image file will be converted into a series of binary data, which is called the *bit-stream*. The decoder then receives the encoded bit-stream and decodes it to form the decoded image. If the total data quantity of the bit-stream is less than the total data quantity of the original image, then this is called image compression [39].

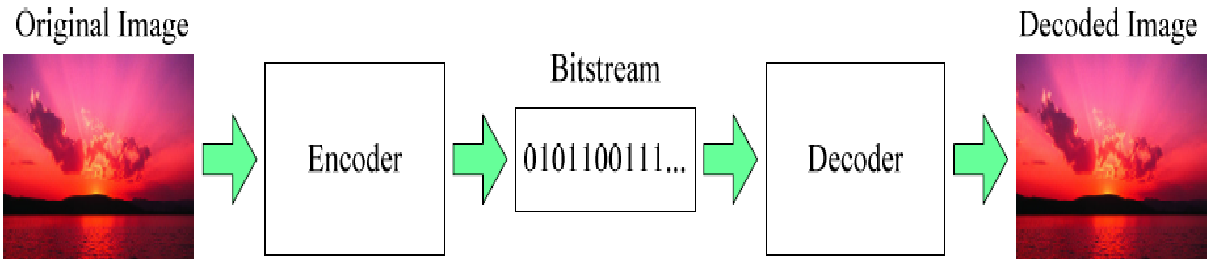

Figure (2.1) The basic image compression model [39].

A typical image encoder consists of three closely connected components:

i. **Transform Coding:** The transform coding is the most important part of the image compression algorithm**.** It is applied to reduce the correlation between the neighboring pixels in the image because the correlation between one pixel and its neighbor pixels is very high and the values of one pixel and its adjacent pixels are very similar [39].

ii. **Quantization:** A quantizer simply reduces the number of bits needed to store transformed coefficients by reducing the precision of those values. Since this is a many-to-one mapping, it is a lossy process and is the main source of compression in an encoder. Quantization can be performed on each individual coefficient, which is known as Scalar Quantization (SQ). Quantization can also be performed on a group of coefficients together, and this is known as Vector Quantization (VQ) [4].

i. **Entropy coding:** An entropy encoder further compresses the quantized values lossless to give better overall compression. It uses a model to accurately determine the probabilities for each quantized value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream [15].

The general encoding architecture of image compression system is shown in Figure (2.2).

Diagram (2.1) The general encoding flow of image compression model [31].

2.3.3 Image Quality Measurements:

The performance of the image compression technique is evaluated by measuring compression efficiency and image quality.

A. Compression Ratio (CR) [30] is measured the compression efficiency, it is defined as the ratio of the compressed image size and the original image size

$$CR = \frac{compressed\ image\ size\ (in\ bits)}{original\ image\ size\ (\ in\ bits)} \qquad (2.2)$$

achieving high CR, will result of less quality of reconstructed images and vice versa.

B. Mean Square Error (MSE) [1]: is used to calculate the error between the original image and the reconstructed image. The calculated MSE value is given below:

$$MSE = \sqrt{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[f(x,y) - g(x,y)]^2} \qquad (2.3)$$

Where M and N are the numbers of rows and columns in the input image matrix respectively and $f(x,y)$ is the pixel value of the original image, $g(x,y)$ is the pixel value of the reconstructed image.

2.3.4 Image compression methods:

In this section the image compression methods that are used in the present work will be described. Also explain the algorithm for implementation these methods using MATLAB.

2.3.4.1 Image compression using PCA, [5]:

PCA is a mathematical formulation used in the reduction of data dimensions which used in image compression algorithm with minimal loss of information.

To show how the PCA compression works, the random variables $X_1 X_2, \dots, X_p$ would be represent the columns of the image matrix. The PCA is begun by transforming the image to that its columns have zero means and unitary variances. This is common, in order to avoid one or the other of the columns having undue influence on the principal components:

$$image\ corrected\ by\ the\ mean = image - mean\ of\ the\ image \quad (2.4)$$

Then finds the covariance matrix of the image by the following formula:

$$covImage = image\ corrected\ by\ the\ mean \times (image\ corrected\ by\ the\ mean)^T \quad (2.5)$$

and computes the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_p$ and the corresponding eigenvectors $\alpha_1, \alpha_2, \dots, \alpha_p$ of the covariance matrix. After that obtained vc which is a matrix with all the eigenvectors (components) of the covariance matrix.

$$vc = \left(\alpha_1, \alpha_2, \dots, \alpha_p\right) \quad (2.6)$$

any components that explain only a small portion of the variation in data for the effect of image compression are discarded. The eliminations have the effect of reducing the quantity of eigenvectors of the characteristics vectors and can produce compressed image with a smaller dimension.

$$Compressed\ Image = vc^T \times (Image - mean)^T \quad (2.7)$$

The reconstructed image is obtained using the equation**:**

$$Reconstructed\ Image = (vc)^T \times Compressed\ Image + mean^T \quad (2.8)$$

2.3.4.2 Image Compression using SVD, [38]:

The SVD is useful in image compression such that it discards elements representing small singular values.

To show how the SVD compression works, let A be a matrix representing a given image after applying SVD on A is refactoring it into product of three matrices U, Σ , and V such that $A = U\Sigma V^T$, then A can be written as:

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T \qquad (2.9)$$

We can then obtain the truncated sum after the first k terms

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T \qquad (2.10)$$

We can choose k to be considerably less than n and still have the image corresponding to A_k very close to the original. The value k represents the rank of the matrix A_k. The magnitude of the smallest nonzero singular value provides a measure of how close A is to a matrix of lower rank. A_k is a compression of the data represented by matrix A, where the amount of storage necessary for the original matrix A was reduced. Small variations on the matrix A may not affect the reconstruction and may well adapt to loss of information. If A_k has rank k by construction, then

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \qquad (2.11)$$

This means that we can represent the image A as a linear combination of the basis images ($u_i v_i^T$). All the basis images are rank one and form an orthonormal basis for image representation. Here, (2.11) is the best rank-k approximation of A . In a sense, the following equation is minimized

$$\|A - A_k\|_2 = \left\|\sum_{i=k+1}^{n} \sigma_i u_i v_i^T\right\|_2$$

$$= \left\| U \begin{bmatrix} 0 & & 0 & \\ 0 & \sigma_{k+1} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & & \dots & \sigma_n \end{bmatrix} V^T \right\|_2 = \sigma_{k+1} \qquad (2.12)$$

The vector space spanned by $v_1, v_2, \dots, v_{k+1}$ has dimension $k+1$. In a $m \times n$ size image, an effective compression technique is needed to represent the image rather than transmitting or storing the entire $m \times n$ matrix. In equation (2.11), it will take $m \times k + n \times k = (m+n) \times k$ words to store u_1 through u_k , and $\sigma_1 v_1$ through $\sigma_k v_k$, from which we can reconstruct A_k. Thus, after compression A_k will be stored using $(m + n) \times k$ words.

2.3.4.3 Image compression using Hybrid Transform (DCT & DWT):

The image compression using Hybrid transform (DCT and DWT) works on the advantages of both DCT and DWT in such a way that size of the image gets reduced to a large extent with higher compression ratio. The first step is transform image by using three level decomposition DWT as shown in Diagram (2.2) where in the first decomposition level the image is decomposed in two parts; approximation part of an image which is low frequency part and another is detailed part of an image which is higher frequency part. Four sub-images are obtained at each level of the decomposition of the image: approximation image (LL1), vertical detailed image (LH1), horizontal detailed image (HL1) and diagonal detailed image (HH1) Where the alphabet L means low pass signal and H means high pass signal also the first alphabet represents the transform in row whereas the second alphabet represents transform in column. Then all the image coefficients are dropped except approximation image (LL1) which is sent for second level decomposition [36]. The process continues for one more level.

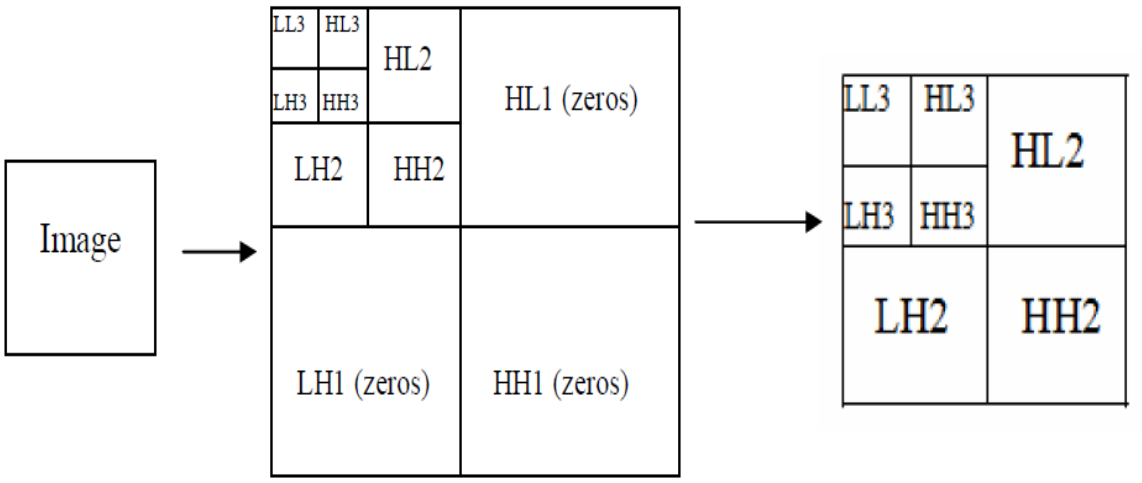

Diagram (2.2) Three level decomposition DWT.

The second step is quantization the detailed image in the second level (LH2, HL2, HH2) and the detailed image in the third level (LH3, HL3, HH3) and eliminate zeros form the matrices after that apply Arithmetic Coding on each of the detailed image to compress it, as shown in Diagram (2.3).

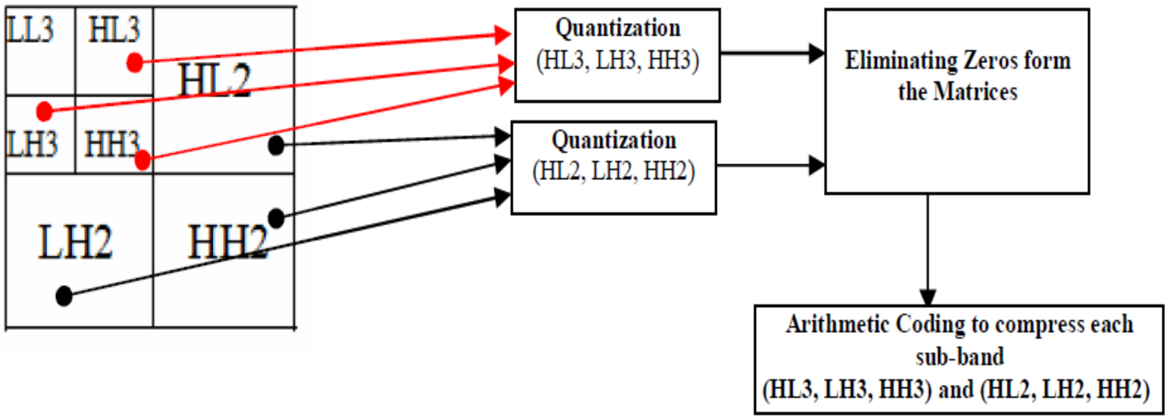

Diagram (2.3) Quantization and eliminate zeros form each sub-band, and then compress each sub-band by Arithmetic Coding.

The third step that shown in Diagram (2.4) includes application the 1-D DCT on the approximation image LL3 then quantization it and convert the matrix to array this process is called T-Matrix coding at the end Arithmetic Coding is applied.

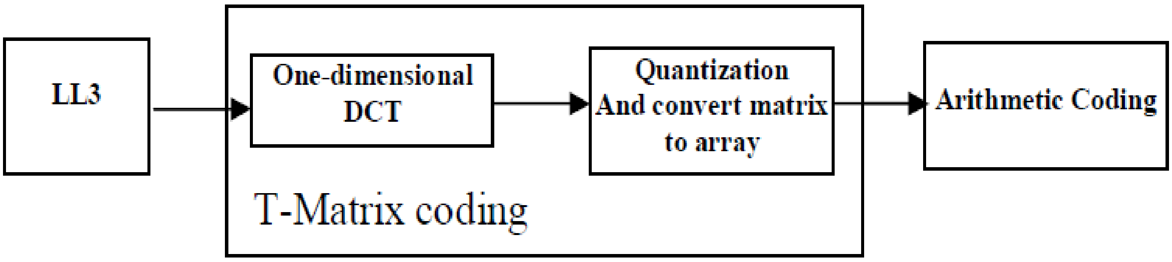

Diagram (2.4) Compress LL3 sub-band by using T-Matrix Coding.

The image is then reconstructed through decompression. Decompression process uses the 1-D IDCT and 2-D IDWT.

2.3.4.4 Image compression using BPNN:

Artificial Neural Networks are gaining immense popularity for their ability to solve complex real-world problems in image compression. In order to achieve compression the image is first passed through the input layer and then through a hidden layer consisting of very small number of neurons. This hidden layer stores the compact features of the image, and hence, the number of neurons in this layer needs to be smaller in order to achieve a higher compression ratio. The compressed image is finally observed after passing through the output layer. This compressed image retains much of the input data, while discarding all the redundant information [34]. The neural network structure for image compression is shown in Figure (2.2).

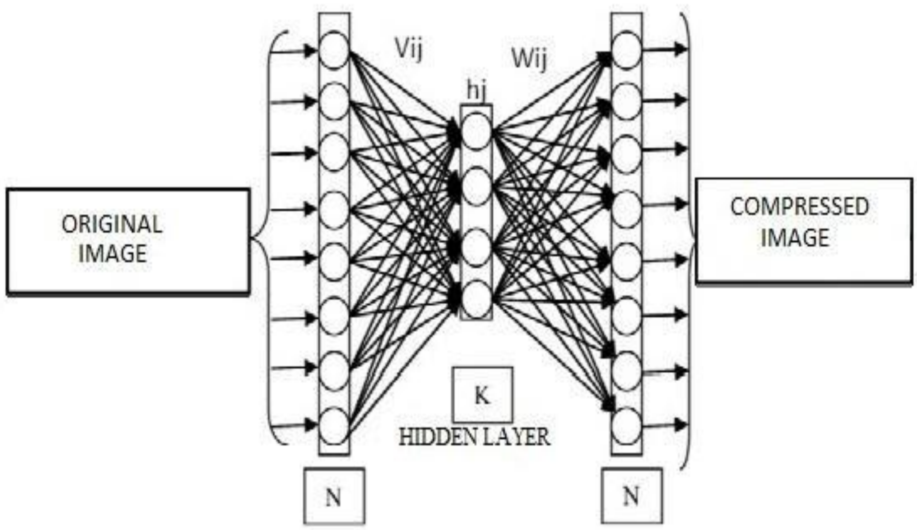

Figure (2.2) Neural Network for Image Compression [34].

Now, the procedure of image compression will explain in a detail. At first dividing the original image into *N* blocks by l pixels and reshaping each one into column vector**.** Arranging all column vectors in a matrix. Let the target matrix equal to the input matrix. Choose a suitable learning algorithm (in our work we use a backpropagation learning), and defining the training parameters: the number of iterations, the number of hidden neurons and the initial conditions. Implement the network with the input matrix and the target matrix to obtain the output matrices of the hidden layer and the output layer. Finally, reshaping each column of these matrices into a block of the desired size and then arrange the blocks to obtain the compressed image, and the reconstructed image respectively [35].

2.4 Wireless Network System:

The traditional ways of networking the world have proven inadequate to meet the challenges posed by our new collective lifestyle. If users must be connected to a network by physical cables, their movement is dramatically reduced. Wireless connectivity, however, poses no such restriction and allows a great deal more free movement on the part of the network user. As a result, wireless technologies are encroaching on the traditional realm of fixed or wired networks [7].

A network is a set of devices connected by communication pathway called link that transfers data from one device to another. A network can be categorized as [6]:

- Local area network (LAN) which is usually privately owned and links the devices in a single office, building, or campus.
- Wide area network (WAN) which provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world.
- Metropolitan area network (MAN) which is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city.

2.4.1 Wireless Network Architectures, [33]:

There are two different network architectures: infrastructure and ad hoc networks.

i. *infrastructure network* is a network in which all stations communicate with each other through an access point (AP) as shown in Figure (2.3). In an infrastructure network, wireless stations can communicate with each other or can communicate through a wired network with other stations.

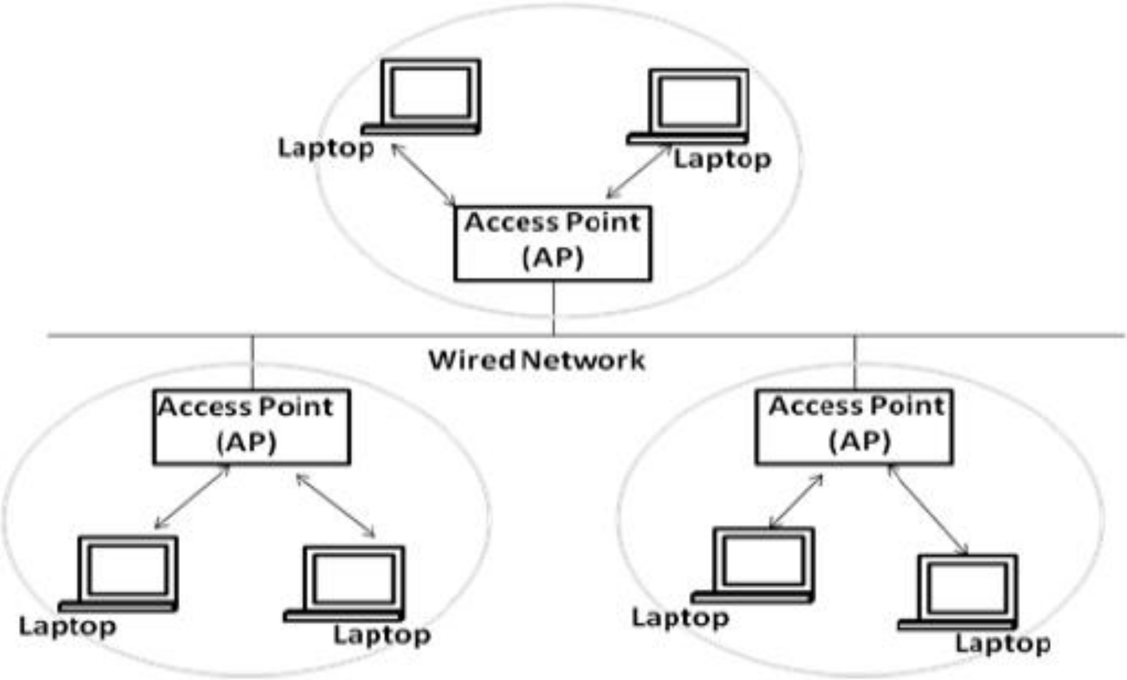

Figure (2.3) infrastructure network [33].

ii. *ad hoc networks* is a network in which stations communicate directly with each other, without the use of an AP as shown in Figure (2.4). Ad hoc networks are useful in cases that temporary network connectivity is required.

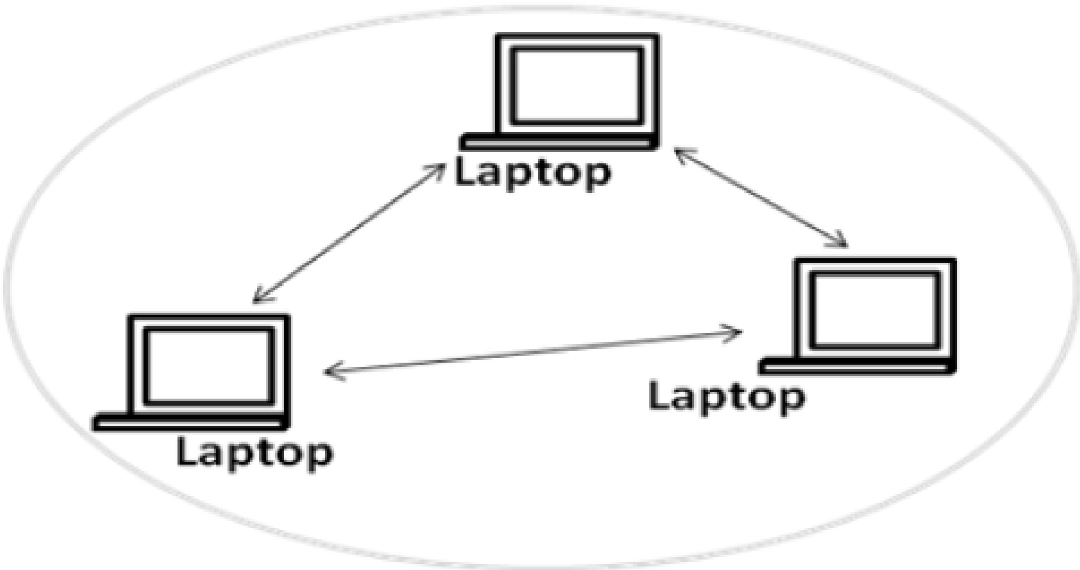

Figure (2.4) ad hoc network [33].

2.4.2 Network communication, [25]:

The main method of network communication is using TCP/IP communication protocol suit that makes use of network sockets to transfer data between devices of the network bi-directionally. In this manner, the computer that is controlling the relay can send commands and shortly thereafter receive the response through the same Socket. In most programming languages, firstly a socket must opened to import the appropriate plug-in, then build the socket object, and connect the socket using the IP Address and Port Number of the target device. The concept of IP addressing, that means that it provides an IP address to every machine on the network in order to deliver data packets, and Port Numbers are basically the entry into a network or device that allow access to its internals from remote communications. IP address and port number play very important role in selecting the final destination of data. After the host is selected by using IP address of it, then the port number defines one of process on that particular host.

2.4.3 Network monitoring, [13]:

With the growth of network applications, the network administrator must have the knowledge of current condition of whole network, so that administrator can manage it responsively and correctly. Administrator therefore need more real-time and powerful network monitoring tools, which can monitor multiple computers and components reside in network simultaneously and can decrease the cost of manpower.

A network monitoring system monitors the network for problems caused by overloaded or crashed servers, network connections or other devices. There are two Monitoring Techniques; Router Based and Non-Router Based. Monitoring functionalities that are built-into the routers themselves and do not require additional installation of hardware or software are referred to as Router Based techniques. Non-Router based techniques require additional hardware and software to be installed and provide greater flexibility.

2.4.4 A wireless distribution system, [37]:

A wireless distribution system (WDS) connects two or more wired or wireless LANs through connects APs wirelessly to establish a large network. WDS networking is classified into:

i. Point-to-point (P2P) network: in this type the WDS uses two APs to implement the wireless bridging of LAN segments 1 and 2 so that LAN segments 1 and 2 can communicate with each other. Figure (2.5) show P2P network.

Figure (2.5) P2P topology [37].

ii. Point-to-multipoint (P2MP) network: as shown in Figure (2.6) AP1 is used as the central AP, and all other APs establish wireless bridges only with AP1. This implements the connection of multiple networks. LAN segments 2, 3, and 4 can only communicate through AP1.

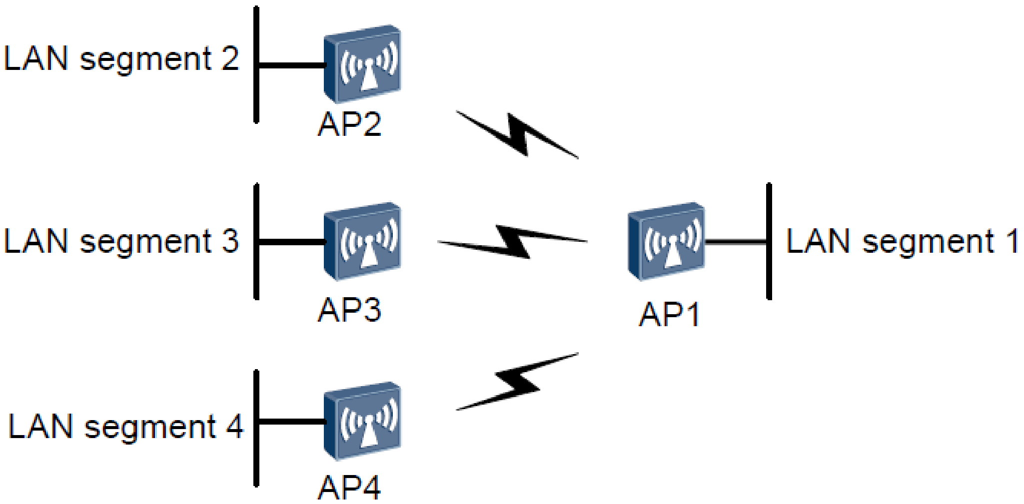

Figure (2.6) P2MP topology [37].

Chapter Three

Design and Implementation Proposal System

Chapter Three
Design and Implementation Proposal System

3.1 Introduction:

This chapter introduce the implementation of image compression methods using MATLAB2015a program and the implementation of the proposal system network with test results.

3.2 Proposal System Network:

The proposal system network is shown in Figure (3.1) which consists of the hardware and software parts.

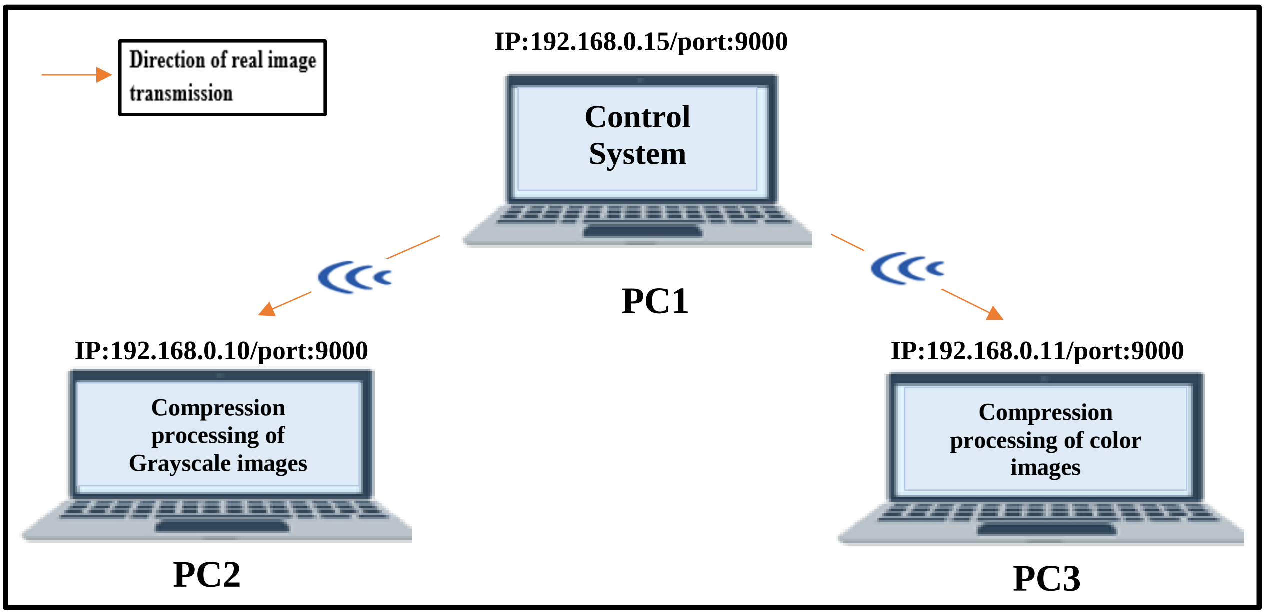

Figure (3.1) Proposal System Network.

The hardware used in the system consists of two parts:

- Wireless communication
- Personal computers

The software used in the system consists of:

- MATLAB program version (R2015a) to compress the images.
- Net Support School Professional software version (10.70.5), which is responsible for network monitoring and controlling.

Networking hardware includes all equipment needed to preform data processing and communications within the network. In Figure (3.1), the proposal network consists of 3 Personal Computers (PCs), each PC communicates with one another through an access point. In the proposed wireless network, PC2 and PC3 are connected wirelessly with PC1.

The overall wireless network is monitored in PC1 (central PC) using Net Support School software program, by setting the central PC (PC1) as a teacher and the rest of network PC's as students as shown in Figures (3.2) and (3.3).

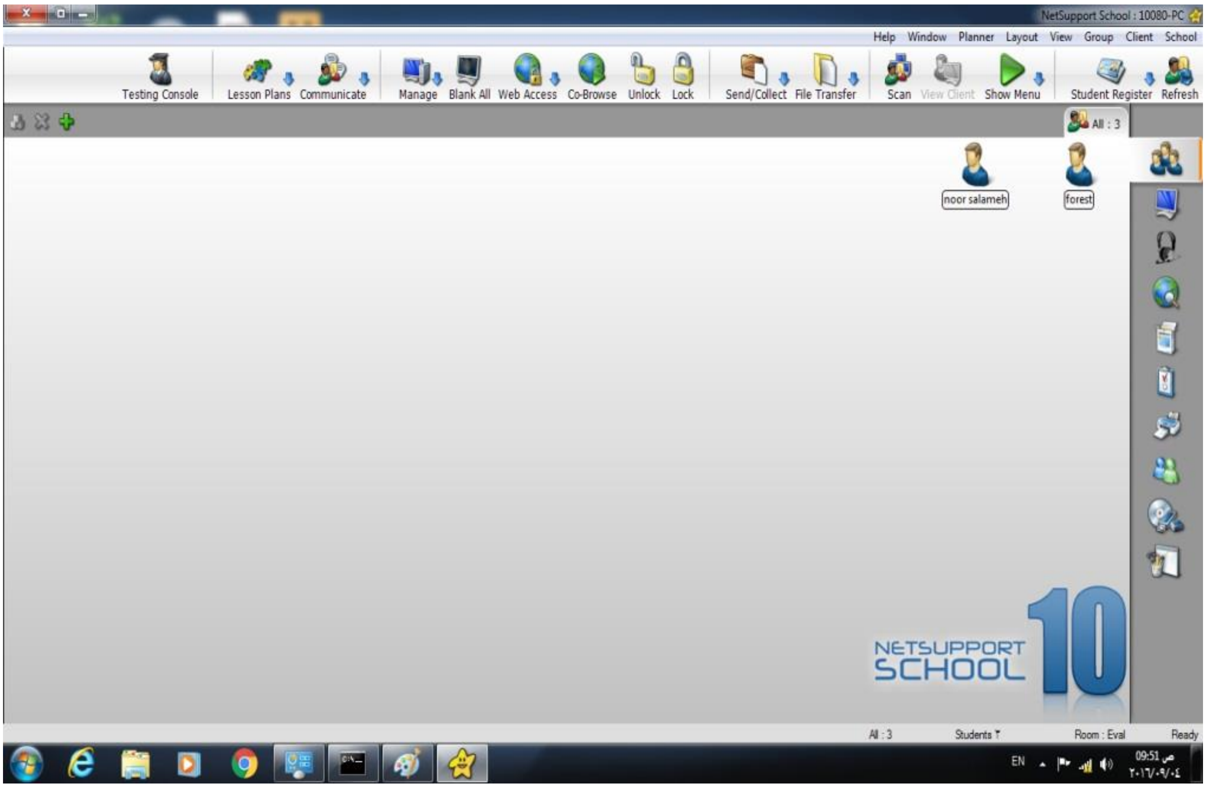

Figure (3.2) Output Net Support School teacher's software.

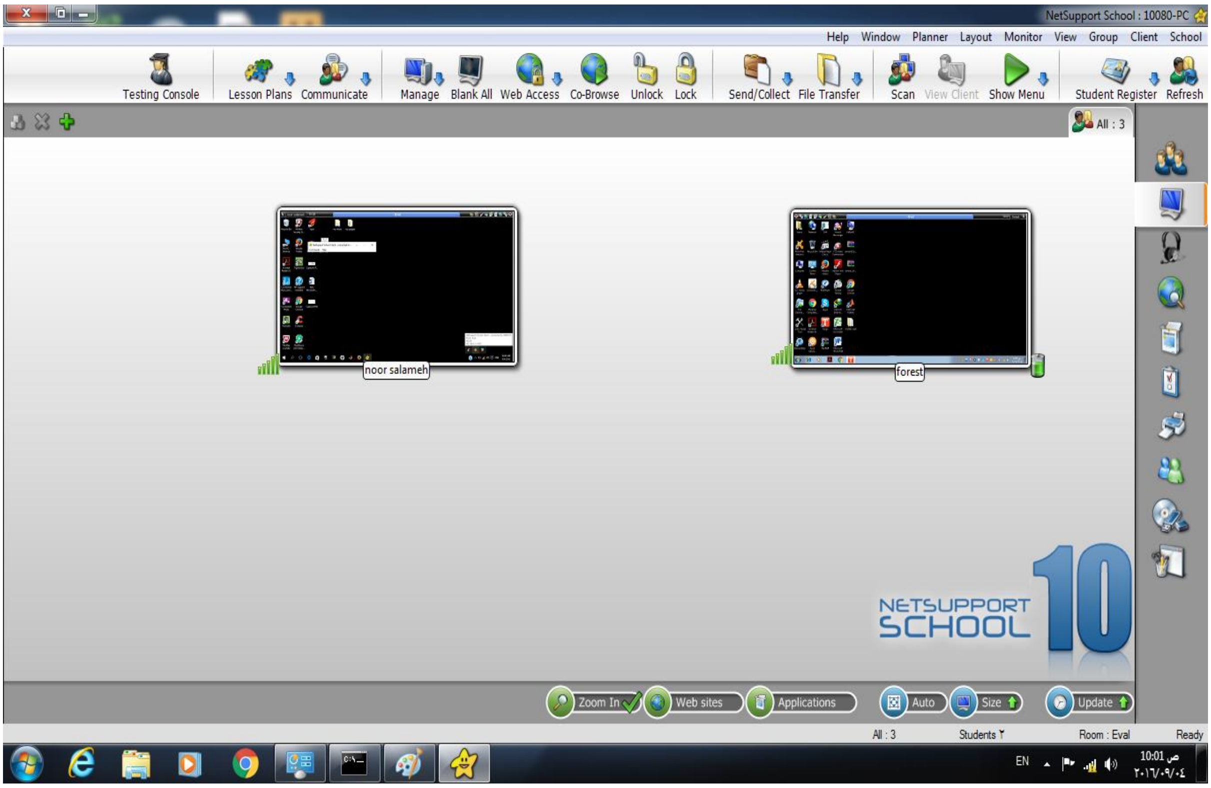

Figure (3.3) Monitoring screen of PC1 that controls PC2 and PC3.

The teacher starts the classroom of the connected students. The teacher PC has a full control on all the available students in the network, so any disconnected student(s) from the network will disappear from the teacher screen. The teacher PC can also have the knowledge that PC's are connected wirelessly.

All of network's PCs transfer data by using Transmission Control Protocol/Internet Protocol (TCP/IP). TCP requires only the destination's IP address and port number for the data to be transferred to that destination.

To make sure that all students are connected properly to the central PC, ping utility only needs the destination's IP address to test a connection. When replay is returned from the specified destination, means that the destination is properly connected into the network as shown in Figure (3.4).

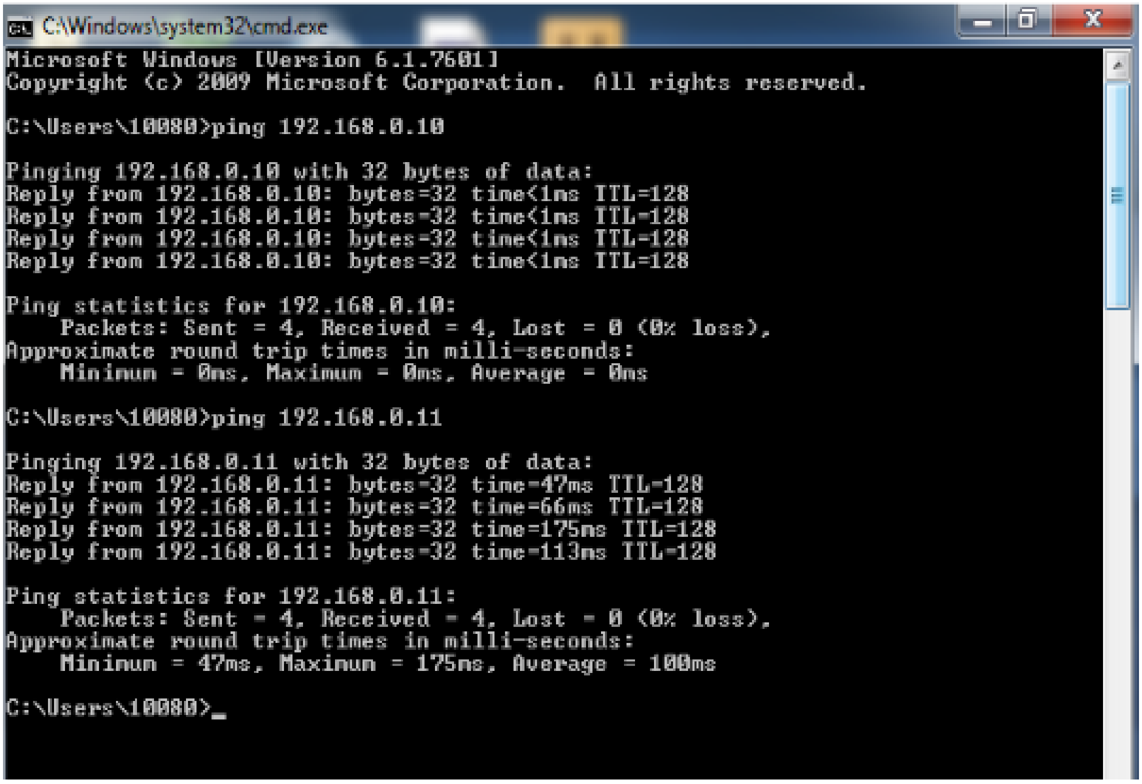

Figure (3.4) Output screen of connecting PCs.

Network monitoring and controlling takes place in PC1 which is the central PC. PC1 sends an image that is selected by the network administrator (the user of central Personal Computer (PC)) to PC2 and PC3.

The test images that used in this work are some standard bit-map (BMP) images. These images are shown in Figure (3.5) which are four 24-bit true color images 256×256.

a. Lena Image **b. Pepper Image** **c. Girl Image** **e. Parrots Image**

Figure (3.5) The original test color images.

Hence PC2 will compress the grayscale image using PCA, SVD, Hybrid (DCT & DWT) and BPNN methods as shown in Figures (3.6), (3.7), (3.8) and (3.9), where the image quality measurements CR and MSE are computed to estimate the quality of the reconstructed grayscale image for each method and the computation time of running compression process as shown in Table (3.1).

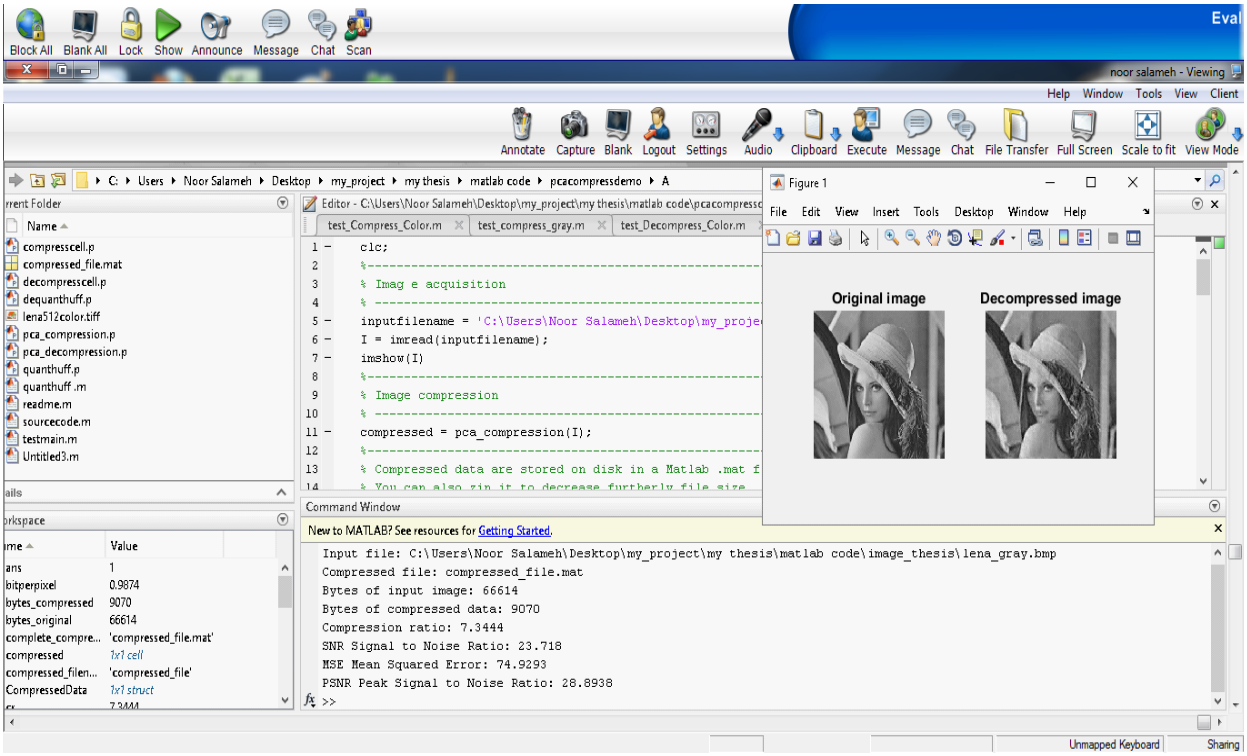

Figure (3.6) Screen shot of Matlab code for Lena image compression using PCA.

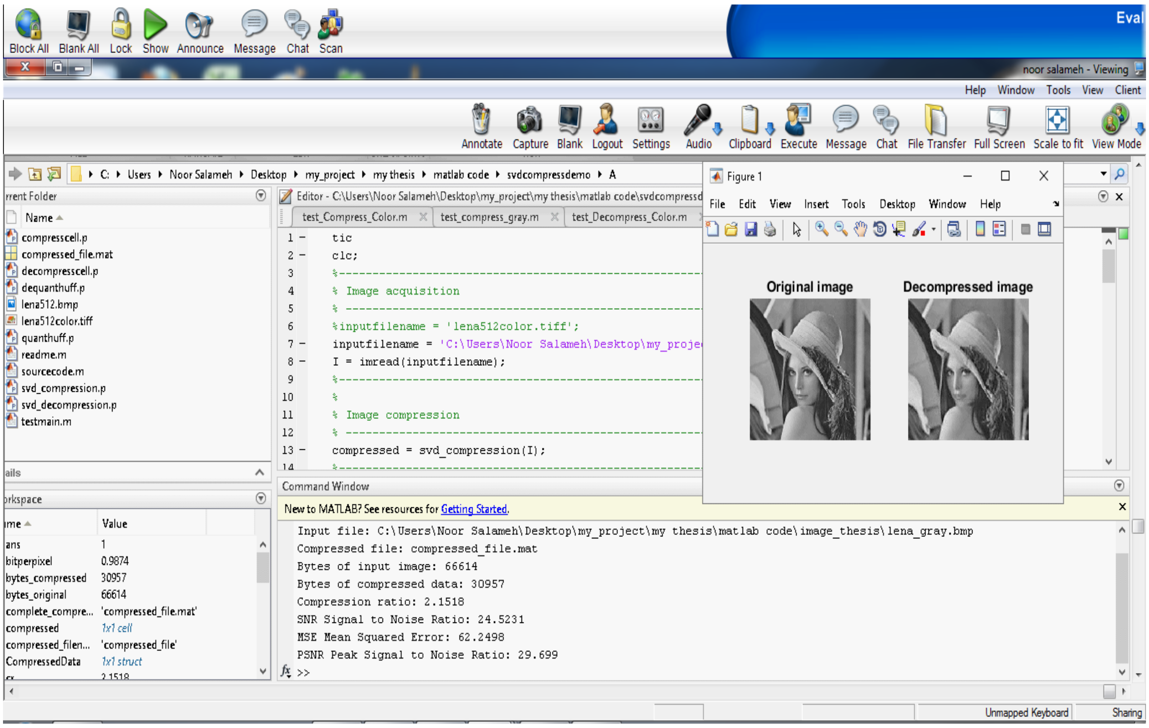

Figure (3.7) Screen shot of Matlab code for Lena image compression using SVD.

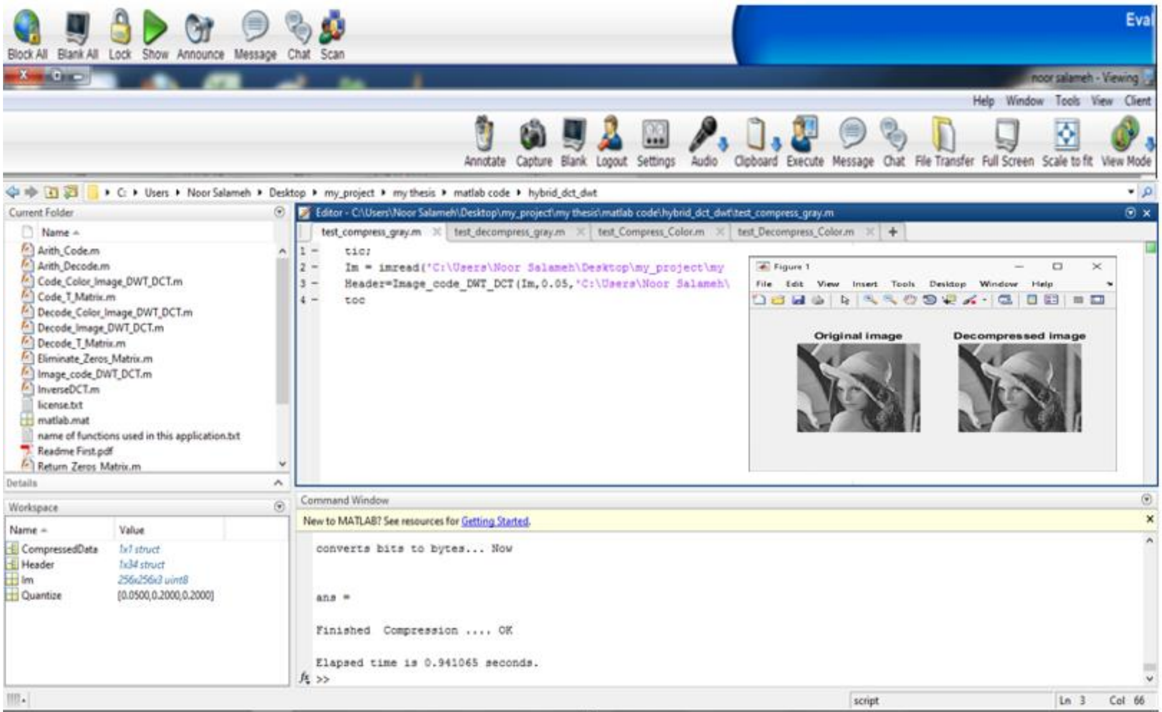

Figure (3.8) Screen shot of Matlab code for Lena image compression using Hybrid (DCT&DWT).

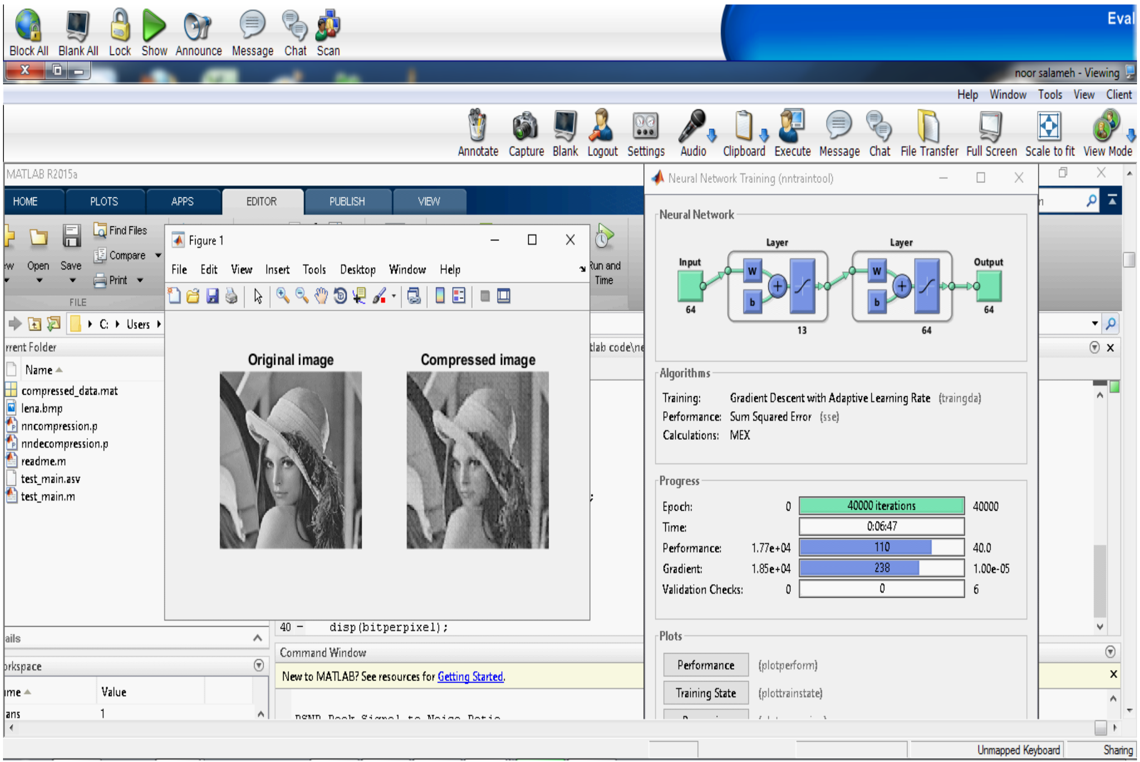

Figure (3.9) Screen shot of Matlab code for Lena image compression using BPNN.

Table (3.1): The results of compression for grayscale images using PCA, SVD, Hybrid (DCT & DWT) & BPNN methods based on C.R and MSE measurements and the computation time of compression process.

Method	Image	MSE	CR	Time (sec)
PCA	**Lena**	74.9293	13.616 %	3.452679
	Pepper	72.8923	14.467 %	4.419505
	Girl	38.6549	13.332 %	3.062416
	Parrots	34.3732	12.632 %	4.245851
SVD	**Lena**	62.2498	46.472 %	37.724621
	Pepper	60.5143	46.75 %	38.505309
	Girl	28.2768	46.24 %	37.236273
	Parrots	19.4707	45.828 %	35.668934
Hybrid (DCT& DWT)	**Lena**	45.688	51.291 %	2.032135
	Pepper	41.7288	52.156 %	2.600004
	Girl	23.8035	47.362 %	2.371254
	Parrots	21.0164	43.751 %	1.664362
BPNN	**Lena**	178.4104	48.084 %	296
	Pepper	113.9579	53.726 %	292
	Girl	92.9816	47.07 %	253
	Parrots	145.405	39.381 %	330

Figure (3.10) shows the reconstruction of all grayscale images after applying PCA, SVD, Hybrid Transform (DCT & DWT) and BPNN methods.

A ***B*** ***C*** ***D***

A ***B*** ***C*** ***D***

A ***B*** ***C*** ***D***

A ***B*** ***C*** ***D***

Figure (3.10) The reconstructed grayscale images of the A- PCA, B- SVD, C- Hybrid (DCT & DWT) and D- BPNN.

Similarly, Hence PC3 will compress the color image using PCA, SVD, and Hybrid (DCT & DWT) methods as shown in Figures (3.11), (3.12) and (3.13), where the image quality measurements CR and MSE are computed to estimate the quality of the reconstructed color image for each methods and the computation time of running compression process as shown in Table (3.2).

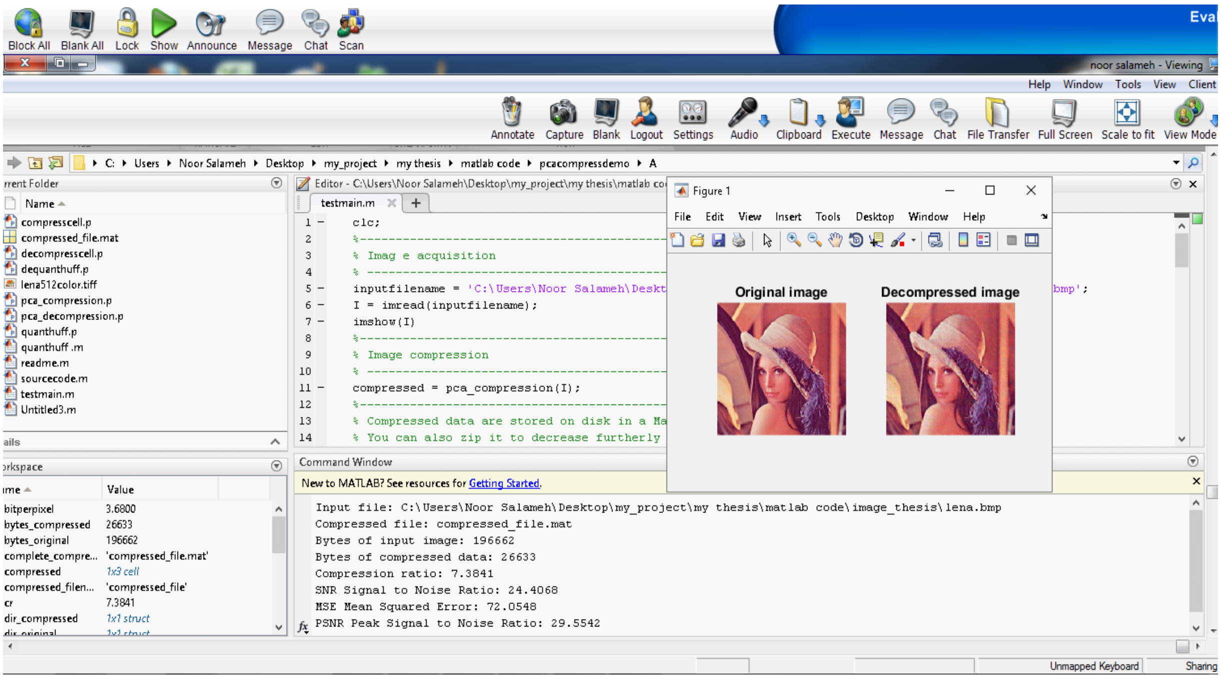

Figure (3.11) Screen shot of Matlab code for Lena image compression using PCA.

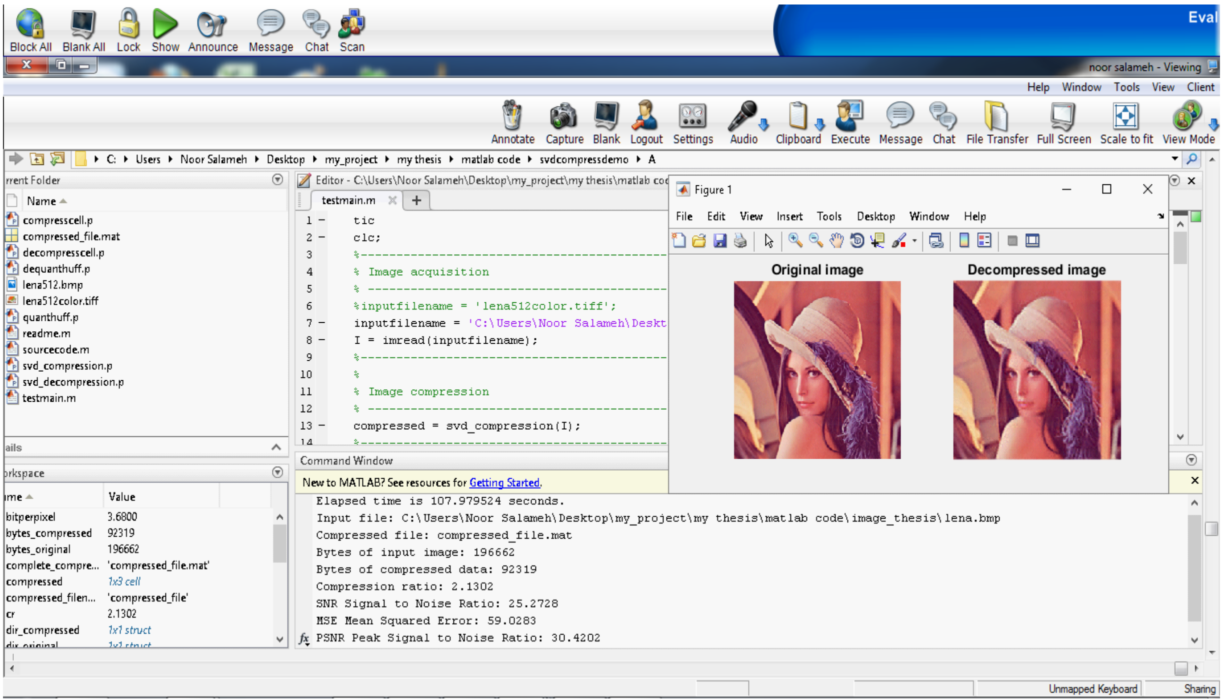

Figure (3.12) Screen shot of Matlab code for Lena image compression using SVD.

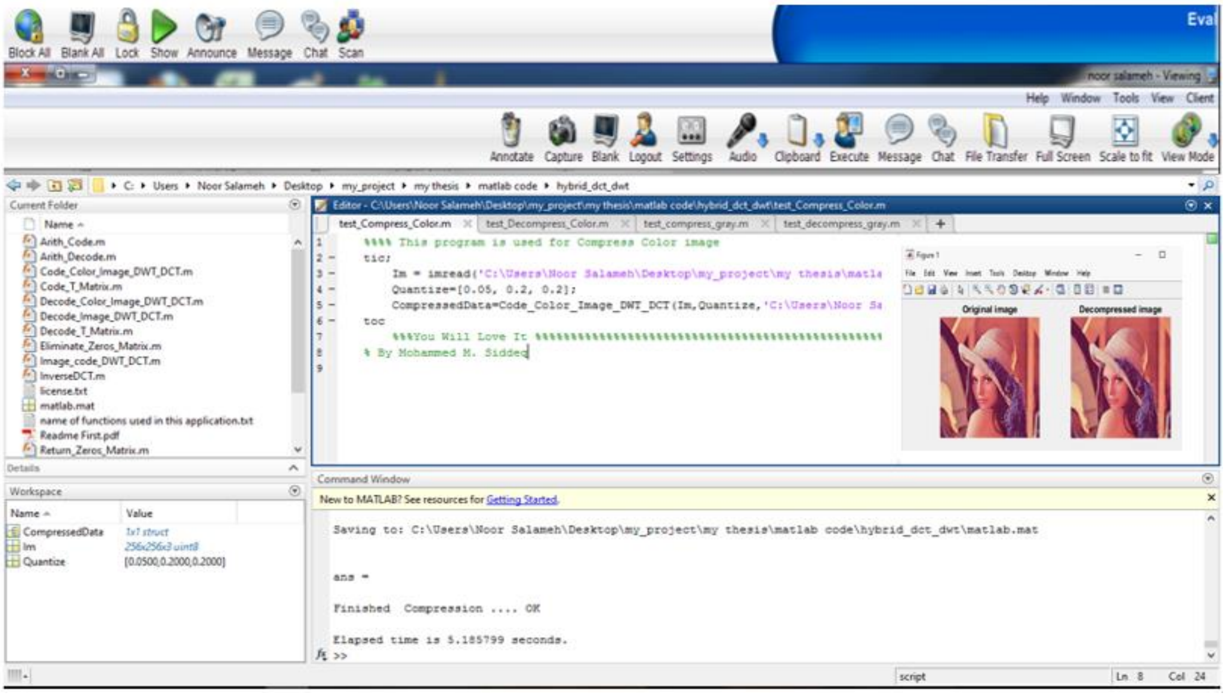

Figure (3.13) Screen shot of Matlab code for Lena image compression using Hybrid (DCT & DWT).

Table (3.2) The results of compression for color images using PCA, SVD and Hybrid (DCT & DWT) methods based on C.R and MSE measurements and the computation time of compression process.

Method	Image	MSE	CR	Time (Sec)
PCA	**Lena**	72.0548	13.543 %	10.602498
	Pepper	76.9453	13.968 %	9.079207
	Girl	42.7997	9.8044 %	8.693298
	Parrots	43.6426	9.489 %	8.874344
SVD	**Lena**	59.0283	46.943 %	98.163820
	Pepper	61.8441	46.727 %	100.990474
	Girl	29.8922	34.493 %	105.768491
	Parrots	22.2397	34.610 %	105.165775
Hybrid (DCT& DWT)	**Lena**	47.6794	53.747 %	5.44747
	Pepper	51.8579	55.243 %	5.495939
	Girl	26.9194	36.754 %	7.988646
	Parrots	23.5958	35.122 %	6.564915

Figure (3.14) shows the reconstruction of all color images after applying PCA, SVD and Hybrid Transform (DCT & DWT) methods.

Figure (3.14) The reconstructed color images of the A- PCA, B- SVD and C- Hybrid (DCT & DWT).

Conclusions and Recommendations

Conclusions and Recommendations

During the implementation of the case studies, a number of conclusions have been drawn based on the practical results obtained from the implemented systems and the following are the most important ones:

- Hybrid (DCT & DWT) technique is the best method to compress the color and grayscale images due to a lower MSE so it gives a best quality for the resulted reconstructed images, also it gives a higher CR and shorter time spend for compression process.
- SVD is a good method to compress color and grayscale images but the time that spend for compression process is longer than the time that Hybrid (DCT & DWT) spent.
- PCA gives a good quality for the reconstructed images but it gives a lower CR between the methods we used.
- BPNN is a weakest method to compress the grayscale images because it gives a worst quality for the reconstructed image despite of a high CR and it takes longest time.

To develop the present implemented work, the following recommendations are put forward:

- Try using another method of image compression to compare with.
- Wireless network transmission can be improved by using another backup router and set up the network as mesh topology incase if one router fail down, the network transmission will not be affected.
- Design new system based on separate compression and decompression processes in order to use security system.

References

References

[1] Acharya, T. and Ray, A. K., "Image Processing Principles and Applications", John Wiley & Sons, Inc., Hoboken, New Jersey, Canada, 2005.

[2] Cherkassky, V., He, X. and Balasubramanian, P., "Wireless Transmission of Image and Video Data", Final Report, Minnesota Department of Transportation Office of Research Services, April, 2002.

[3] Chowdhury, M. M. and Khatun, A., “Image Compression Using Discrete Wavelet Transform”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No. 1, July, PP. 327- 330, 2012.

[4] Dhawan, S., "A Review of Image Compression and Comparison of its Algorithms", International Journal of Electronics & Communication Technology, Vol. 2, Issue. 1, March, PP. 22-26, 2011.

[5] Espírito Santo, R., "Principal Component Analysis applied to digital image compression", Einstein, Vol. 10, No. 2, June, PP. 135-139, 2012.

[6] Forouzan, B. A., "Data Communication and Networking”, 4th ed., McGraw-Hill Companies, Inc., New York, 2007.

[7] Gast, M. S., "802.11 Wireless Networks the Definitive Guide”, O'Reilly, USA, 2005.

[8] Gonzalez, R. C., Woods, R. E. and Eddins, S. L.,” Digital Image Processing Using MATLAB”, 2nd ed., Gatesmark, LLC, 2009.

[9] Gupta, M. and Garg, A. K., “Analysis of Image Compression Algorithm Using DCT”, International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 1, Jan-Feb, PP. 515-521, 2012.

[10] Jain, A. K., Mao, J. and Mohiuddin K. M., “Artificial Neural Networks: A Tutorial “, IEEE, March, PP. 31- 44, 1996.

[11] Jaiswal, N., “Image Compression Using Back Propagation Neural Network”, International Journal of Scientific Engineering and Research (IJSER), Vo. 3, Issue 5, May, PP. 61-64, 2015.

[12] Jolliffe, I.T, "Principle Component Analysis", 2nd ed., Springer- Verlag, New York, Inc., 2002.

[13] Joshi, M. and Gohel, C., "Agent Base Network Traffic Monitoring", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 2, Issue 5, May, PP. 1799- 1803, 2013.

[14] Kahu, S. and Rahate, R., "Image Compression using Singular Value Decomposition", International Journal of Advancements in Research & Technology, Vo.2, Issue 8, August, pp. 244-248, 2013.

[15] Kapoor, P. and Patyal, S., "DCT Image Compression for Color Images", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 2, Issue 10, October, PP. 3247 – 3252, 2014.

[16] Khayam, S. A., "The Discrete Cosine Transform (DCT): Theory and Application", ECE 802 – 602: Information Theory and Coding, March 10th 2003.

[17] Kshirsagar, G., Amaria, R., Chandel, M. and Kakade, Sh., "Stock Prediction using Artificial Neural Networks", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 6, Issue 3, March, PP. 626 - 630, 2016.

[18] Leon, S. J., "Linear Algebra with Applications", 8th Edition, Pearson Education, Inc., 2010.

[19] Le-Rademacher, J. G., "Principal Component Analysis for Interval-Valued and Histogram-Valued Data and Likelihood Functions and Some Maximum Likelihood Estimators for Symbolic Data", Ph.D. Thesis, University of Georgia, Athens, Georgia, 2008.

[20] Lin, P.-Y., "An Introduction to Wavelet Transform", Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, ROC, 2007.

[21] Mitchell, T. M., "Machine Learning", McGraw-Hill Science, 1997.

[22] Mohammad, M.-B and Mostafa, M.-B, "Color Image Compression using PCA", International Journal of Computer Applications, Vol. 111, No. 5, February, PP. 16- 19, 2015.

[23] Olkkonen, J., "Discrete Wavelet Transforms -Theory and Application", InTech, 2011.

[24] Panda, S .S., Prasad, M.S.R.S, Prasad, MNM and Naidu, Ch. S., "Image Compression Using Back Propagation Neural Network", International Journal of Engineering Science & Advanced Technology (IJESAT), Vol. 2, Issue 1, Jan-Feb, PP. 74- 78, 2012.

[25] Parziale, L., Britt, D. T., Davis, C., Forrester, J., Liu, W., Matthews, C. and Rosselot, N., "TCP/IP Tutorial and Technical Overview", 8th ed., IBM: Internal Technical and Support Organization, 2006.

[26] Patel, B. K. and Agrawal, S., "Image Compression Techniques Using Artificial Neural Network", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 2, No. 10, October, PP. 2725- 2729, 2013.

[27] Poswal, V. and Priyanka, "Analysis of Image Compression Techniques using DCT", International Journal of Electronics and Computer Science Engineering (IJECSE), Vol. 1, No. 3, PP.1727-1733, 2012.

[28] Rajeswari, K., Kavitha, K. and Boopathi Raja, G., "Comparison of Image Compression Techniques using DWT, DAST and LZW with PCA", International Journal for Scientific Research & Development (IJSRD), Vol. 3, Issue 3, 2015.

[29] Sachdeva, S. and Kaur, R., "A Review on Digital Image Compression Techniques", International Journal on Recent and Innovation Trends in Computing and Communication IJRITCC, Vol. 2, Issue 7, July, PP. 1909 – 1913, 2014.

[30] Salmon, D., "Data Compression", 3rd ed., Springer-Verlag New York, Inc., 2004.

[31] Sandhu, G. K., Mann, G. S. and Kaur, R., "Benefit and security issues in wireless technologies: Wi-fi and WiMax", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Issue 4, June, pp. 976- 982, 2013.

[32] Senthilkumaran, N. and Suguna, J., "Neural Network Technique for Lossless Image Compression Using X-Ray Images", International Journal of Computer and Electrical Engineering, Vol.3, No.1, February, PP.17- 23, 2011.

[33] Singh, R. K. and Jain, A.K., "Research Issues in Wireless Networks", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 4, April, PP.115-119, 2012.

[34] Somanathan, A. M. and Kalaichelvi, V., "An Intelligent Technique for Image Compression", International Journal of Recent Development in Engineering and Technology, Vol. 2, Issue 4, June, PP. 1-6, 2014.

[35] Srikala, P. and Faruq, S. U., "Neural Network Based Image Compression With Lifting Scheme and RLC", IJRET: International Journal of Research in Engineering and Technology, Vol. 01. Issue 01, September, PP. 13- 19, 2012.

[36] Ul Ain, A. and Khaitan, S., "Image Compression Using Hybrid (DCT+DWT) Technique - A Comparative Study", International Journal on Recent and Innovation Trends in Computing and Communication, Vo. 4, Issue 5, May, PP. 105-111, 2016.

[37] "WDS Technology White Paper", Issue 1, Huawei Technologies Co., Ltd., 2013.

[38] Workalemahu, T., "Singular Value Decomposition in Image Noise Filtering and Reconstruction", Master thesis, Department of Mathematics and Statistics, College of Arts and Sciences, Georgia State University, 2008.

[39] Wei, W.-Y., "An Introduction to Image Compression", Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, ROC., 2008

المستخلص

الهدف من هذه الرساله هو تصميم وتنفيذ نظام شبكه لاسلكيه يتكون من حاسوب شخصي رئيسي وحاسوبان ثانويان يتصلون مع بعضهم بواسطة جهاز التوجيه حيث يقوم الحاسوب الرئيسي بالسيطره والتحكم ببقية الحواسيب الاخرى للشبكه. جميع حواسيب الشبكه تتصل مع بعضها بواسطة (TCP\IP) المتحكم بالحاسوب الرئيسي للشبكه يقوم باختيار الصوره المطلوبه ويقوم بارسالها لبقية الحواسيب بحيث تقوم واحده بضغط الصور الغير ملونه والحاسوب الاخر يقوم بضغط الصور الملونه باستخدام الطرائق التاليه: تحليل المكونات الرئيسيه ، وتحليل القيمة المفرده، والطريقه الهجينه (منفصلة تحويل جيب التمام ومنفصلة تحويل المويجات) و الشبكة العصبية ذي الانتشار الخلفي. حيث تتم المقارنه بين طرائق الضغط هذه تتم بالاعتماد على بعض المقاييس كنسبة الضغط ومقدار الخطأ بين الصوره الاصليه والصوره المغضوطه لتوضيح دقة الصوره وبالاعتماد على الوقت المستغرق في عملية الضغط . واعطت الطريقه الهجينه افضل النتائج لان جودة الصوره المضغوطه التي تعطيها عاليه ولها نسبة ضغط عاليه وتستغرق عملية الضغط وقت قصير.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم
قسم الرياضيات وتطبيقات الحاسوب

تصميم وتنفيذ نظام شبكه لاسلكيه لضغظ وفك ضغط الصوره

رسالة
مقدمه الى مجلس كلية العلوم – جامعة النهرين
وهي جزء من متطلبات نيل درجة ماجستير علوم
في الرياضيات

من قبل

نور سلامه شحده
(بكالوريوس علوم، جامعة النهرين، ٢٠١٤)

اشراف

أ. د. علي عبد الحافظ ابراهيم

تشرين الأول ٢٠١٦ ذو الحجه ١٤٣٧