Spring 2005

Image Compression by Vector Quantization of DCT Coefficients Using a Self-Organizing Neural Network

Kirankumar Boyapati
Old Dominion University

IMAGE COMPRESSION BY VECTOR QUANTIZATION OF DCT COEFFICIENTS USING A SELF-ORGANIZING NEURAL NETWORK

by

Kirankumar Boyapati
B.E. August 2002, Madras University, India.

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
May 2005

Approved by:

Dr. Vijayan K. Asari (Director)

Dr. Stephen A. Zahorian (Member)

Dr Min Song (Member)

ABSTRACT

IMAGE COMPRESSION BY VECTOR QUANTIZATION OF DCT COEFFICIENTS USING A SELF-ORGANIZING NEURAL NETWORK

Kirankumar Boyapati
Old Dominion University, 2005
Director: Dr. Vijayan K. Asari

Image compression is used to remove different types of redundancies and irrelevant information in an image so that the space required for storing the image is minimized. Conventional transform based techniques remove the redundancies between adjacent pixels and irrelevant information in the image but fail to remove the redundancies between different parts of the image. There are techniques like vector quantization, which take care of the redundant information in different parts of the image, but fail to remove the irrelevant information in the image. In this thesis a new technique for lossy digital image compression is proposed which is a combination of both transform based compression and vector quantization. The algorithm presented in this thesis not only removes the redundancy between neighboring pixels and the irrelevant information present in the image but also the redundancy between different parts of the image. The compression ratio attained by this algorithm is better compared to both transform based and vector quantization based compression techniques.

The new image compression technique first uses Discrete Cosine Transform (DCT) to separate the high frequency information. This high frequency information is removed completely or quantized heavily depending on the compression required. A self-organizing neural network based technique in combination with vector quantization is

used to detect different parts of the image that are similar. These blocks of the image, which are similar, are classified and represented using a single entry in the codebook. A first order predictor method instead of a conventional zero order predictor has been used to encode the index. The first order predictor takes advantage of the fact that the change of gradient in an image is smooth. The change of gradient is assumed to follow the direction with minimum change in gradient and any change in the value of the current block from the predicted value is stored in the index. The results of the experiments conducted using this algorithm show an increase in performance by approximately 20% to 80% (in terms of compression) when compared to the JPEG standard. When compared to JPEG 2000, the proposed algorithm yields better compression at higher rates. Research work is in progress to improve the performance of the algorithm by incorporating an optimum quantization matrix and an adaptively computed vigilance value for the self-organizing neural network.

This thesis is dedicated to

My family members for their love and motivation,

my advisor Dr. Vijayan K. Asari whose continued support, patience and esteemed guidance helped me complete this thesis successfully,

and my friends who have stood by me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Image compression is becoming more and more important with the rapid increase in the amount of digital image used for various applications. Storing images without compressing requires huge amount of digital memory, which is not only costly but also makes applications like downloading loads of images from satellites very difficult if not impossible. Image compression not only makes storing of digital image easier but also makes retrieval of images from anywhere around the world using the internet fast and easy. The images are encoded at the sender before transmission and the compressed data is transmitted, which takes less time. This encoded data is decoded at the receiver side to get back the reconstructed image. Hence compression methods are required, which perform better compression with a minimum loss in clarity of the image.

Image compression can be achieved by saving the image more efficiently, after removing different types of redundancies existing in the image. Redundancies like correlation between the neighboring pixels can be removed to store only the required information for reducing the size of the digital image. General compression methods can be applied to compress images but the compression obtained is not optimal. Compression of an image is significantly different from other types of data compression as images have certain statistical properties, which can be exploited by designing encoders specifically to attain better compression.

The reference model for this work is *IEEE Transactions on Image Processing*.

1.1. Lossless and Lossy Compression Modes

Image compression can be divided into two types, lossless and lossy. Lossless image compression is used in applications where exact reproduction of the original image is required whereas lossy mode is used in applications where a reasonable reproduction of the original image is good enough.

1.1.1 Lossless Compression

This mode of compression is used in applications like compression of medical images, where loss of any data from the image is unacceptable. In lossless compression the redundancies in the image are removed to attain compression without any loss of information. In this compression mode care should be taken to avoid the rounding errors inherent in digital computers. In lossless compression the reconstructed image is an exact replica of the original image. The quantization error induced in an image when converted from analog to digital form is lost forever and cannot be retrieved.

1.1.2. Lossy Compression

The compression ratio of lossless methods are not high enough for image compression, especially when the distribution of the pixel values is relatively flat. Image compression is different from data compression as in the case of an image a good approximation to the original image is enough for most purposes unlike data compression where data loss is generally not acceptable. Lossy image compression is used in applications where loss of information is not fatal, like storage of natural images etc. Lossy compression algorithms usually have an adjustable parameter to control compression with the quality of output image decreasing with increase in the compression. Compression ratios for images compressed using lossy image compression are much higher than lossless image compression with the image being indistinguishable from the original. Higher compression can be obtained in lossy image compression as not only are the redundancies in the image removed but also irrelevant information is also eliminated.

(a) Original image.

(b) Reconstructed image with lossless compression. (c) Reconstructed image with lossy compression.

Fig. 1.1: Illustration of the effects of lossless and lossy compression.

Fig. 1.1 (b) and 1.1 (c) show the reconstructed images compressed from the original image in fig. 1.1 (a) using both lossless and lossy modes of compression respectively. The compression attained by lossless compression is 2:1 whereas the compression attained using lossy image compression is 15:1 with the images being similar to each other. The compression ratio attained with a reconstructed image of reasonable quality varies from image to image depending on the redundancies and irrelevant data present in the image. The quality of the reconstructed image is determined by two metrics, Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR), which give a reasonable understanding of the quality of the reconstructed image.

1.2. Redundancy

Redundancy is the difference between the shortest way information could be coded and the data actually used to represent the information. Redundancies are basically of three types: interpixel redundancy, coding redundancy and visual redundancy. All of the image compression techniques take advantage of one or more of these types of redundancies inherent in the original image to compress the image effectively.

1.2.1. Interpixel Redundancy

Images usually have a smooth variation in gradient and hence a pixel value can be predicted using the value of the neighboring pixels as they are correlated. This type of redundancy is predominantly found in natural images as natural images have a very slow change in gradient and is called interpixel redundancy. Transform-based techniques are used to remove this redundancy.

1.2.2. Coding Redundancy

Instead of representing all the pixels with the same number of bits, information theory can be used to code images such that the pixels with higher probability of occurrence are represented with fewer bits and the pixels with lower probability use more. This reduces the total number of bits used to encode the image when the information of an image is concentrated near a certain value, which usually is the case with natural images. Efficiency of the coding depends on the entropy of the source. Entropy of an image is a quantity that is used to quantify the amount of information, that must be encoded by the compression algorithm. An image with lower entropy can be compressed to a relatively smaller size than an image with higher entropy. Images with higher entropy have a greater contrast from one pixel to the next as the data are random and unpredictable and hence cannot be compressed as much as low entropy images. This type of redundancy is called coding redundancy, which can be removed by using coding techniques like Huffman coding.

1.2.3. Visual Redundancy

Visual redundancy is the information in the image, like high frequency information, which has minimum or no impact on the human eye. This property can be used to attain higher compression by removing the higher frequency components. Transform based methods can be used to distinguish high frequency components from the low frequency components. Visual redundancy can be used in lossy image compression but not in lossless image compression, as information is lost in this process.

1.3. Proposed Technique

Compression techniques using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are widely used. Transforms used in image compression convert source data in spatial domain into frequency domain. Conversion into frequency domain helps in distinguishing the low frequency components from the higher frequency components; making it easier to either eliminate the high frequency components completely or quantize them heavily as they have lesser impact on human visual system. DCT based techniques are used in existing Joint Photographic Experts Group (JPEG) standard whereas the new JPEG 2000 standard is based on wavelet transforms. Wavelet transforms have an advantage over DCT as they separate smooth variations and details in the image by decomposition of the image to separate the information in the image into different frequency sub-bands. Wavelet transform better represents the time and frequency information simultaneously compared to transforms like DCT where either time or frequency can be provided but not both.

Compression using DCT and DWT take advantage of the correlation between neighboring pixels to reduce the redundancy and the irrelevant high frequency components from the original image. These transforms fail to take advantage of another type of redundancy, which exists between different parts of the image that may be similar. There are other techniques like vector quantization that consider this type of redundancy but fail to remove the irrelevant high frequency information from the image.

Vector quantization techniques first divide the image into small blocks and a codebook is formed which should be as small as possible and at the same time be able to give a reasonable approximation of all the blocks in the image. The compressed image contains the code table and index of which code in the code table is nearest to the current block. At the decoder each block in the image is replaced by its corresponding code word from the codebook.

A neural network based approach is proposed which combines the advantage of both the compression techniques. This algorithm removes the redundancy between adjacent pixels of the image and the redundancy between different parts of the image. This also removes the high frequency components from the image, which are irrelevant to attain better compression ratios. The computations required by this approach are greater than the computations required for compression techniques using DCT or vector quantization but attain higher compression ratios without affecting the image quality. The increase in compression is worth the additional computation especially with the growth of computation power far exceeding the growth in the field of computer networks.

The remainder of the thesis is organized as follows: Chapter II discusses the basic concepts of digital image and image compression in general. In Chapter III various transform techniques with their advantages and disadvantages are discussed. Chapter IV contains the proposed method for digital image compression with detailed description of the algorithm. Chapter V summarizes the results of the study with graphical analysis of the compression results, which include subjective and objective qualities of the reconstructed image followed by conclusion along with suggestions for future research in Chapter VI.

CHAPTER II

REVIEW OF EXISTING TECHNIQUES AND ALGORITHMS

In the course of the research work conducted to aid this thesis, many books and published articles were read and referred to. This chapter documents the resources and research materials, which were very helpful in the research work done for this thesis.

2.1. Representation of Digital Image

Digital images can be created from an analog image by taking samples, which are an approximation of the analog image at each pixel [1]. At a given point (x, y) it is possible to represent the intensity of an image with finite accuracy using a finite number of bits. Thus if a set of points (x, y) is chosen for representation, and the brightness of each point is represented by a number of bits, we have a matrix represented as B (x, y) with a finite amount of binary data [2]. The number of bits required to represent all the possible representations of the intensity depend on how close the digital image should be to its analog counter part. It is not feasible to represent a digital image with all the details in the analog domain.

The interval between adjacent samples determines the resolution of the sampled signal. The resolution of the digital image increases with the increase in number of samples taken in a given area and hence resolution provides the upper limit on the frequencies that can be represented in a given digital image. Resolution beyond a point is of no real use as high frequency components add details like fine spatial information and small amplitude variations, which have a very little or no impact on human eye. Each sample in a digital image is called a pixel or a pel and the number of bits required to represent each pixel to the required level of details is called precision. The precision of a digital image should be high enough to give the appearance of continuity to the observer and as small as possible to reduce the size of the image. Digital images normally are represented using 8 bits per pixel (bpp) for monochrome images and 24 bpp for color images.

Digital images are classified into two categories, gray scale images and color images. The gray scale images, as the name signifies, contain different possible gray levels depending on the precision of the image. These images are also called as monochrome images. Color images are more frequently used and can have different representations. According to the trichromatic theory, ideally at least three arrays of samples, with each array representing each component are required to properly represent a color image. This means digital representation of a color image is a three dimensional array with the third dimension showing the number of components used to represent an image. Color images can be represented by three basic color components, Red, Green and Blue (RGB). All the resultant colors in an image can be obtained by varying the scale of these components in a relative scale of 0 to 1. Color CRT monitors and most computer graphics systems use the RGB color model. The RGB model simplifies the design of computer graphics systems but is not ideal for all applications.

In RGB the color components are highly correlated, which makes it difficult for many image-processing techniques. Image processing techniques like image compression concentrate more on the intensity component of an image [3]. These processes are easily implemented using the Hue, Saturation and Intensity (HSI) color model. HSI is represented by components that are closely related to the criteria used to describe color perception by human eye, like brightness, hue and saturation. Brightness describes the intensity of the light (revealing if it is white, gray or black) and this can be related to the luminance of the source. Hue describes what color is present which can be related to the purity or narrowness of the spectral distribution.

Representations in which one component is luminance and the other two components are related to hue and saturation are called luminance-chrominance representations. Luminance provides gray scale version of the image and the chrominance components provide the extra information required for converting grayscale image to a color image. Luminance-chrominance representations are very useful for image compression as the components closely represent human perception of the color [4]. Each component or layer of the image can be viewed as a single channel image, which can be analyzed independently from the others. All three components can be treated individually and compressed. Change in a component does not drastically affect the image. This is not the case with RGB samples where if two channels are fixed, human visual perception is very sensitive to even small change in the value of the third channel [5]. Thus even though RGB is the most common storage format for images, Luminance-Chrominance format is preferred for image compression.

Image compression methods discussed here concentrate mainly on gray scale images as even color images use the same algorithms, once for each component. The quantization matrix is different for different components depending on the importance of each component on the human visual system.

2.2. Image Compression

Image compression means reducing the number of bits required to describe an image. The image is compressed at the encoder and the output from encoder is the compressed data. This compressed data is used for storage and transmission and the image is reconstructed from compressed data at the decoder. The purpose of image compression is to represent an image using the least number of bits to save storage costs or transmission time. Effective compression can be achieved by making an approximation of the original image rather than reproducing it exactly. The greater the compression required, the lossier the image is likely to be as loss of information increases with compression. In lossy image compression care is taken to make sure that the information lost during compression has the least impact on the human eye resulting in a reconstructed image, which looks closer to the original image even with some of the information being lost. In some applications where loss of information is not acceptable, lossless image compression is used to compress the image as far as possible without any loss of information. In lossless image compression technique, removal of the redundant information alone is responsible for image compression whereas lossy image compression also takes advantage of the irrelevant information for better compression.

2.3. Coding

General image coding can be classified into pulse code modulation or predictive coding and sub-band or transform coding. A combination of both these coding techniques is used in most of the compression algorithms to attain better compression [6]. A brief description of the coding techniques is given in this section.

2.3.1. Predictive Coding

In predictive coding an attempt is made to predict the pel to be encoded. The prediction is made using the values of the previously encoded pixels and only the predictive error is stored [7,8]. This approach can be made adaptive by changing the prediction to suit local picture statistics. Compression efficiency can be increased by not encoding the error if the error in prediction is below a particular threshold. Another possibility is to delay the encoding of a pel until the future trend of the signal can be observed, and then coding the pel to take advantage of the trend. This process of coding is called delayed coding.

2.3.2. Transform Coding

In transform coding instead of coding the image as discrete values with intensity at sampling points, an alternative representation is made first by linearly transforming the pels into blocks of data called coefficients which represent the amplitude of the signal at different frequencies rather than intensity at each pixel. These coefficients are then quantized depending on the frequencies they represent. Several transformations have been used to convert information in spatial domain into corresponding frequency domain. DCT is one of the most popular transforms used for image compression. Two-dimensional transforms are used as they represent the frequencies in both x-axis and y-axis accurately. With sub-band coding, the image is passed through a bank of band-pass filters, thus decomposing the waveform into a number of sub-band pictures, which can then be sub-sampled and coded. Sub-band coding, using wavelets, is used in the new JPEG standard. Various transforms used for image compression along with their properties, advantages and disadvantages are discussed in the next chapter.

2.4. Image Compression Standards

The Joint Photographic Experts Group has developed an international standard for general purpose, color and still-image compression. Standards are mandatory for image compression; without standards we probably will have to download a decoder with every image we download from the internet, as decoder should have some *a priori* knowledge of the encoder used to code the image. In this section we discuss some of the existing image compression standards.

2.4.1. JPEG Standard

JPEG standard is the existing standard for image compression which uses transform based coding [9,10]. The model of JPEG compression encoder is shown in fig. 2.1 and the decoder is shown in fig. 2.2.

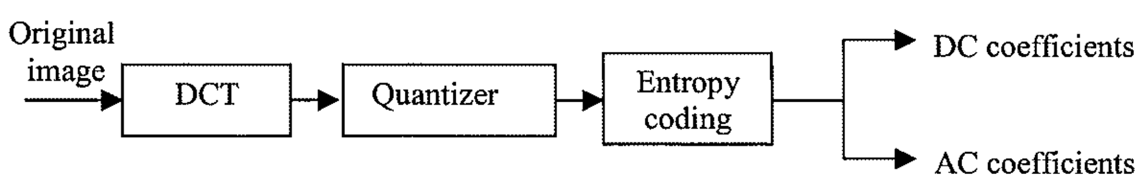


Fig. 2.1: Block schematic representation of the JPEG encoder.

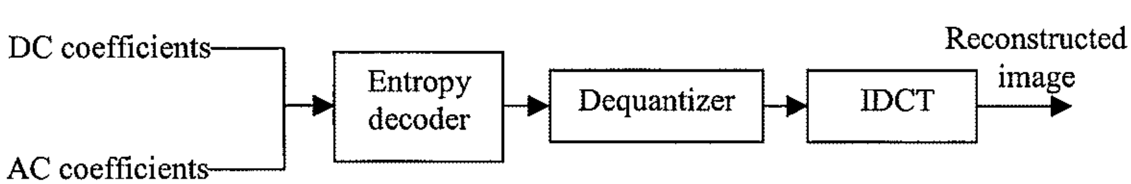


Fig. 2.2: Block schematic representation of the JPEG decoder.

If the color image to be compressed is represented in RGB format, it has to be converted to luminance-chrominance components before compressing it. Luminance-chrominance format closely represents the functionality of human eye and hence is preferred in image compression. These luminance, chrominance components are then treated as completely different images and compressed independently at the encoder and quantized using different quantization matrices which are developed considering the effect of each component on human eye.

Level shift is applied to the image; that is 128 is subtracted from all the pixels to change the range of coefficients from 0 - 255 to −128 - 127. Although not necessary from a mathematical point of view, level shift reduces the precision requirements for the DCT computation. The level shift brings the DC values into a similar signed integer range as the AC values, so the whole DCT calculation is more consistent. The input image is first divided into smaller blocks of same size (8 × 8) to decrease the computation required for computing the transform coefficients. The size of the block is selected as a tradeoff between compression efficiency and computations. The sides are padded with zeros to make the image as a multiple of (8 × 8) blocks. This transform is lossless as it is reversible but introduces small errors due to computational rounding in digital computers.

$$C(i,j) = a(i,j)\sum_{x=0}^{7}\sum_{y=0}^{7} V(y,x)\cos[(2y+1)i\pi/16]\cos[(2x+1)j\pi/16] \quad (2.1)$$

where a[0,0….7], a[0…..7,0] = 1/8, a[1…7, 1…7] = 2/8

The cosine transform changes the image, which is in spatial domain into frequency domain. Two-dimensional DCT is calculated using the formula in eqn. (2.1). Two-dimensional DCT can also be calculated by performing one-dimensional DCT in both column and row wise, which reduces the hardware required for implementation [11]. After the DCT is calculated the horizontal frequency of the basic function increases from top to bottom and the vertical frequency increases from left to right. The result obtained from the cosine transform is an 8×8 matrix where the top left coefficient gives the amplitude of the signal at the lowest frequency which is nothing but an average of the intensities of all 64 pixels and the bottom right coefficient represents the amplitude at the highest frequency. The lowest frequency component is known as the DC coefficient and the remaining 63 coefficients are called AC coefficients [12,13]. As there is low correlation between the coefficients, each coefficient can be treated separately without affecting its neighbors by much. The algorithm calculates DCT separately for every 8×8 block whose result is a matrix of the same size representing weights at different frequencies.

After the DCT, a uniform quantizer is used to quantize each of the 64 resulting DCT coefficients. This quantization process causes loss of image information with different step sizes resulting in different amounts of information lost with larger quantization steps resulting in greater reduction in quality. The quantization table, which is a matrix of quantization steps, is selected such that the quantization step is large for higher frequencies and smaller for lower frequencies to take advantage of the relation between frequency and human visual system. Another important property of the DCT coefficient is that it tends to be highly correlated among adjacent image blocks. This is because the average gray level of adjacent image blocks is likely to be similar as the gradient in most of the images varies smoothly. To improve image quality and reduce bit rate, the DC coefficient is differentially encoded. This means that, using a raster ordering of the blocks. The difference between the current and previous DC coefficients is coded. Most of the remaining 63 AC coefficients have a high probability of being zero after quantization because the higher frequency coefficients usually have lower energy and are also heavily quantized.

These coefficients are then arranged such that zeros are grouped together, forming long sequences of zeros by ordering them in zigzag scan pattern. A Zigzag pattern is used to arrange the coefficients roughly from lower frequency to higher frequency. Different quantization tables are used for each component in the case of color images. The human eye is more sensitive to differences in intensity than to the differences in color. Hence the luminance component is quantized with a matrix with smaller values whereas the other two components are highly quantized. In lossless image compression, the quantization matrix has all the values equivalent to one so that there is no loss of information in quantization phase. Each block is encoded as an integer sequence of bytes, starting with the differential DC coefficient and then the zigzag ordered AC coefficients. As most of the AC coefficients will be zero, zero run length coding is used to effectively code continuous strings of zeros. The baseline encoding uses Huffman coding to reduce the average number of bits. Huffman coding represents data such that the codes with most occurrences require least number of bits and codes with least number of occurrences require more number of bits.

$$V(y,x)=\sum_{x=0}^{7}\sum_{y=0}^{7}C(i,j)a(i,j)\cos[(2y+1)i\pi/16]\cos[(2x+1)j\pi/16] \qquad (2.2)$$

where a[0,0....7], a[0.....7,0] = 1/8, a[1...7, 1...7] = 2/8;

At the decompressor entire process is reversed to get back the reconstructed image. Inverse DCT, which is calculated at the decompressor side, can be calculated by using eqn. (2.2). One of the disadvantages of the conventional JPEG is that at very high rates of compression blocking artifacts are visible.

2.4.2. JPEG 2000

JPEG-2000 is an emerging standard for still image compression. The JPEG 2000 image compression system has a rate-distortion advantage over the original JPEG [14]. It is designed to support features like extraction of image into different resolutions from the same JPEG 2000 compressed bit stream. This allows an application to manipulate or transmit only the essential information for any target device from any JPEG 2000 compressed source data. It also supports various properties like rotating and cropping the image in compressed domain itself. Region of interest is another property supported by JPEG 2000, which means coding the image such that information in the required parts of the image are left intact or with very little loss while other parts of the image which are not of much importance are heavily quantized to obtain better compression rates without loss of information in important regions. The basic ingredients of the JPEG 2000 algorithm are the discrete wavelet transform and the binary arithmetic bit plane coding.

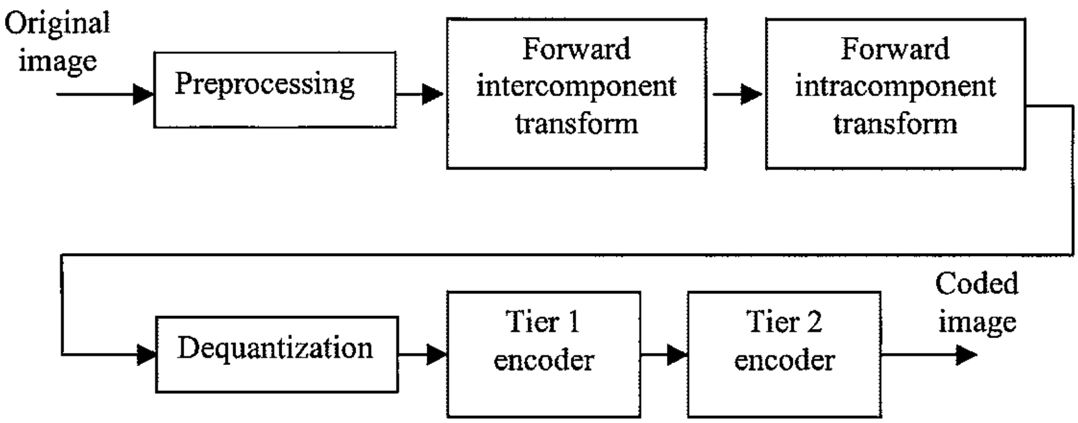


Fig. 2.3: Block schematic representation of the JPEG 2000 encoder.

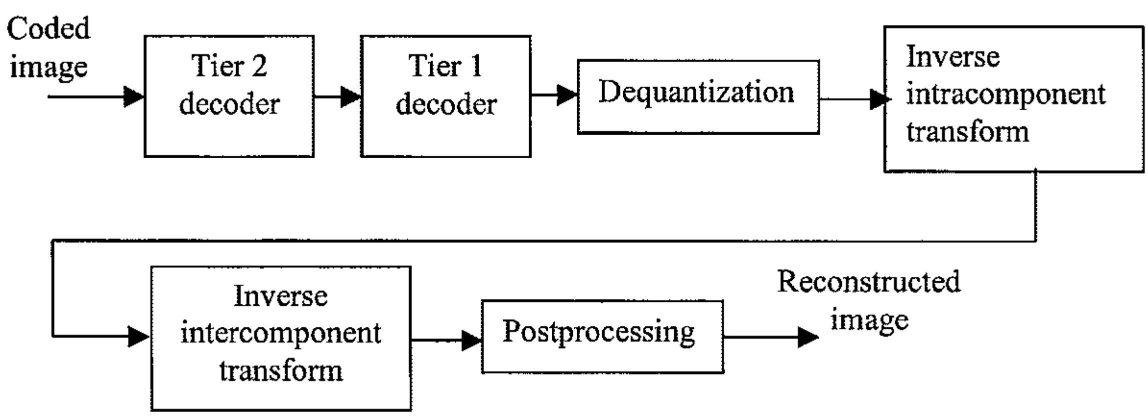


Fig. 2.4: Block schematic representation of the JPEG 2000 decoder.

The encoder for JPEG 2000 standard is shown in fig. 2.3. The preprocessing phase by performing level shift ensures that the input sampled data is approximately centered along zero. The next step is to divide the image into rectangular, non-overlapping blocks. Arbitrary tile sizes are allowed, up to and including the entire image (i.e., no blocks). The size of the tile does not have much of an effect on the computation. Devices with shortage of random access memory use smaller tile sizes. Components with different sub sampling factors are tiled with respect to a high-resolution grid, which ensures spatial consistency of the resulting tile-components. Each tile of a component must be of the same size, with the exception of tiles around the border of the image.

The forward intercomponent transform converts the image from RGB to YUV format using different methods of transformation for lossy and lossless modes. Transform used in lossy mode is called Irreversible Component Transform (ICT) and the transform used in lossless mode is called Reversible Component Transform (RCT). Care is taken such that there is no rounding error in the case of RCT. These transforms can be applied only if there are at least three components with first three components having same resolution.

Intracomponent transform operates on individual components and wavelet transforms are used in this phase [15]. The component is split into numerous frequency bands called sub-bands. Due to the statistical properties of these sub-bands, data can be coded efficiently when uniform maximally decimated filter banks (UMFDB) are applied on an image recursively, applying two-dimensional UMFDB to the low pass sub-band signal obtained at the previous level [16]. Two-dimensional UMFDB is calculated by applying one-dimensional UMFDB vertically and horizontally.

Intracomponent phase consists of two types of wavelet transforms, one for lossy and the other for lossless compression. Lossy image compression uses Daubechies 9/7-wavelet filter and the lossless image compression uses reversible 5/3 wavelet filters [17]. When a two-dimensional wavelet transform is applied to image, four resultant images are obtained. The first image contains the lower frequency components from both vertical and horizontal axis, the second and third images have high frequency components of one axis and low frequency components of the other, and the final image contains the higher frequency components of both the axis [18]. The transform is recursively applied on the image with lower frequencies in both the axis to obtain images each of which contain data, which is of a particular band of frequency.

Two filtering modes are supported by the JPEG 2000 standard namely convolution based and lifting based. Lifting based filtering is used to reduce the computation required in filtering [19,21]. Quantization phase allows greater compression to be achieved by representing transform coefficients with only minimum precision required to obtain a desired level of image quality.

After all the sub-bands have been partitioned into code blocks, each of the code blocks is independently coded using a bit plane coder. Bit plane consists of three coding passes: significance, refinement and cleanup out of which significance and refinement are implemented using arithmetic coder or using lazy mode but cleanup pass always employs arithmetic coding. Lazy mode is a mode in which only four MSB planes use arithmetic coding while all others are raw coded increasing the speed of operation with reduced code efficiency.

First coding pass for each bit plane is the significant pass. Sample is predicted to be significant if any of the eight connected neighbors have already been found to be significant. If arithmetic coding is used, binary symbol conveying significance information is coded using one of the nine contexts. Sign prediction is obtained by using significance and sign information of the four connecting neighbors. In refinement pass, if a sample was found to be significant in previous bit plane, the next most significant bit sampled is conveyed. If arithmetic coding is used each refinement symbol is coded using one of three contexts. Context employed is based on the second MSB position being refined and significance of eight connected neighbors. Cleanup pass is used to convey significant and sign information for those samples that have not yet been found significant and are predicted to remain insignificant during the process of current bit plane. In the tier-two phase, the coding pass information is packaged into data units called packets to obtain compressed image. The data is stored into packages such that at the decoder side the information can be accessed as per requirement. The decoder side processes the coded data in reverse order to get the reconstructed image.

2.5. Vector Quantization

Other types of image compression techniques like vector quantization are also used. Vector quantization is a lossy compression method, which classifies data depending on their similarity so that all similar blocks are clustered to take advantage of redundancy between different parts of the image to attain higher rates of image compression. The image is first divided into smaller blocks and they are classified such that similar blocks are grouped together. Neural network techniques are used to classify similar blocks in an image and a single block in the codebook is used to represent all the blocks classified to be similar. This method is lossy as it considers vectors that are different to be same if the difference between them is below the threshold, hence losing the data that is different for the two blocks. If compression required is higher, the threshold limit is increased. The result of which blocks that are a bit farther from each other also have their differences below the threshold level and hence are assumed to be similar reducing the quality of image drastically. This method in its primitive form is not suitable for image compression, as it does not take advantage of the relation between frequency and human eye. The visual quality of image decreases drastically with increase in compression as some of the information lost during classification is from low frequencies, which have a high impact on human visual system.

A vector quantizer is composed of two operations: the first is the encoder, and the second is the decoder. The encoder takes an input vector and outputs the index of the codeword that offers the lowest distortion along with the code book that contains all the code words required to reconstruct the image [22,30]. The lowest distortion is found by using neural network based classifiers, which calculate the distance between the input vector and each codeword in the codebook and decide if the two input vectors are similar depending on the tolerance limit [31 – 35]. Once the closest codeword is found, the index of that codeword is sent through a channel (the channel could be a computer storage, communications channel, and so on). When the encoder receives the index of the codeword, it replaces the index with the associated codeword. The size of the codebook, N is to be determined based on the output size required. N code words are selected initially and they are assumed to be the codebook. Neural network based methods are used to find the input vectors that are similar to the vectors in the code book and the code word in the code book is modified to be the average of both the code words. This process is called training the neural network and it is repeated until the changes in the code words of the codebook are negligible.

2.6. Neural Networks

An artificial neural network is an information processing system built by the way human brain processes the information [36 – 42]. Neural networks are mainly classified into two categories: recurrent neural networks and non-recurrent neural networks. Recurrent neural networks are the networks, which have a feedback so that the output depends not only on the inputs but also the previous outputs of the network. The non-recurrent neural networks do not have feed back and the output is dependent entirely on the input and the previous outputs do not have any impact on the current output. The neural network based method proposed here uses Fuzzy Adaptive Resonance Theory (Fuzzy-ART) to classify the input vectors into code words which together make a codebook. Fuzzy logic is used along with neural network based on ART to decrease the computation required for classification.

CHAPTER III

TRANSFORMS USED IN IMAGE COMPRESSION

A transform is a reversible linear function in which kernel describes a set of complete, orthonormal and discrete basic functions. Transforms are basically used in image compression to convert the image, which is in spatial domain into frequency domain. By decomposing the image into a set of waveforms each with a particular frequency, we can separate the information that the eye is highly sensitive. Higher rates of compression can be obtained without much visual distortion in the reconstructed image by making sure that the information that is lost during image compression is from the frequencies, which have least impact on human eye. This is done by heavily quantizing the high frequency components and the lower frequency components are quantized with smaller steps or not quantized at all depending on the compression required. The final reconstructed image is as close to the original image as possible, when seen by the human eye or even visually lossless for given rate of compression.

The goal of the transform is to decorrelate the original signal, and this decorrelation redistributes energy among only a small set of transform coefficients, which helps in discarding many coefficients after quantization, requiring fewer number of bits for transmission or storage. This property of transform, which represents the energy of signal in lesser number of bits, is called energy compaction. As information can be effectively represented in smaller number of coefficients energy compaction is a required property for transforms used in image compression.

Transforms should also be completely reversible; that is the original image should be reconstructed from transformed coefficients without any loss in information. Out of the image compression techniques available, transform coding is the most preferred method as energy distribution varies from image to image which makes compression in the spatial domain difficult. Images usually tend to compact their energy in the frequency domain making compression in the frequency domain easier and effective. Transform coding is simply the compression of the images in the frequency domain where transform coefficients, which represent weights of different frequencies in the signal, are encoded to maximize compression. For lossless compression, the coefficients must not allow for the loss of any information and proper care must be taken to avoid rounding error inherent in digital computers. Various transforms used in image compression with their advantages and disadvantages are discussed.

3.1. Hartley Transform

Fourier transform is difficult to apply in image compression as it depends on complex numbers. This problem can be avoided by using a similar transform known as the Hartley transform. This transform works entirely in the real number domain and the lossy compression techniques used do not considerably degrade the image quality even for reasonably high rates of compression. The Hartley transform results in real values for real inputs making it easier to implement when compared to Fourier transform with all the advantages of Fourier transform. This integral shares some of the properties of Fourier transform, but multiplies the kernel with the difference of cosine transform and sine transform instead of cosine transform and the imaginary part of sine transform. Equation of Hartley transform in analog domain is given in eqn. (3.1).

$$H(f) = \int_{-\infty}^{\infty} h(t)(\cos(ft) + \sin(ft))dt \tag{3.1}$$

Equation for the discrete Hartley transform is given in eqn. (3.2). The Hartley transform is its own inverse, which means the eqn. (3.2) can be used to get back the original matrix from the transform coefficients.

$$H(a) = \frac{1}{\sqrt{N}}\left[\sum_{n=0}^{N-1} a_n \cos\left(\frac{2\pi kn}{N}\right) - \sin\left(\frac{2\pi kn}{N}\right)\right] \tag{3.2}$$

The computations required for computing Hartley transform are higher compared to other transforms like DCT, as Hartley transform uses both sine and cosine components instead of just cosine component used in DCT. Computation time required can be reduced by using fast Hartley transforms. Another advantage of Hartley transform is that the same hardware designed for encoding can be used for decoding reducing the hardware requirement.

3.2. Discrete Cosine Transform (DCT)

DCT is one of the most frequently used transforms for image compression. This transform is used in the JPEG standard for image compression in both lossy and lossless modes and Moving Picture Experts Group (MPEG) for video compression. DCT can provide a good approximation of the decomposition by converting the digital image, which is in spatial domain into frequency domain. Cosine transform is suitable to image compression as it is closest to the optimal Karhunen-Loeve Transform (KLT) which by its very definition minimizes error induced by coefficient quantization for signal decorrelation. For highly correlated image data, DCT energy compaction is close to optimal. Energy compaction helps minimize statistical dependency and still gets back the original image with the minimum possible distortion. Cosine transform decomposes each input block into a series of waveforms, each with a particular spatial frequency. The DCT transformation can be viewed as the process of finding the corresponding weight for each waveform so that the sum of all the waveforms scaled by the corresponding weights yields the reconstructed version of the original. Two-dimensional DCT is calculated using the formula in eqn. (2.1) and two-dimensional inverse DCT is calculated using the formula in eqn. (2.2).

DCT is preferred in image compression over transforms like Discrete Fourier Transform (DFT) as it mirrors signal in both directions fixing phase of cosines as the result of which imaginary components get cancelled resulting in the coefficients being real compared to complex coefficients like that of Fourier transform. This is a major advantage in image compression, as storage space and computation required for real numbers is much reduced when compared to complex coefficients. Another advantage of DCT compared to DFT is blocking artifact. Original image is segmented into smaller blocks to decrease the computations required for calculating the transform coefficients. Due to the property of cyclic assumption underlying the Fourier transform it assumes pixels along opposite borders of the block to be similar. When this is not the case, reconstruction of the image from only a partial number of coefficients available after quantization will result in erroneous values along the borders. This phenomenon called Gibbs phenomenon, tends to make the block boundaries distorted and visible. Another drawback of the DFT is the blocking artifact. As a rule original image is segmented into smaller blocks to reduce computation, which results in the image distorted when borders of two blocks are joined at the decoder. Since DCT assumes mirror symmetry, the discontinuities are not that severe reducing the block artifacts significantly.

DCT computations can be performed with fast algorithms that require fewer operations than the computations performed directly from these equations. The computation for the forward and inverse DCT is similar and hence the same hardware computation unit can be used for both forward and the inverse DCT computations. Another advantage of DCT is separability, which means the two dimensional DCT can be calculated by first performing one dimensional DCT with respect to row followed by one dimensional DCT with respect to the column. This row-column approach considerably simplifies the hardware requirements at the expense of a slight increase in computation. Fast algorithms are desirable for both hardware and a software implementation viewpoint as they run parallelly and thus can be efficiently implemented on parallel architectures.

The human eye can detect coarse details in the luminance image better than it can detect fine details. So if the fine details are mathematically sorted out from a luminance block and removed to save storage space, the image will still have an acceptable appearance. Running the 8×8 block of luminance values through a frequency domain transform results in an 8×8 block of frequency coefficients. The low frequency coefficients correspond to the coarse details of the block, while the high frequency coefficients correspond to the fine details of the block. High compression ratios can be obtained, at the expense of fine details, by removing high frequency coefficients. Transforms convert signal that is a function of time into a set of amplitudes for frequency components. The DCT relocates the highest energies to the upper left corner of the image. The lesser energy or information is relocated into other areas, which allows for lossy compression of image data by determining which information can be thrown away without compromising the visual quality of the image. For typical images, many of the DCT coefficients have values close to zero. These coefficients can be discarded without any effect on the quality of the reconstructed image.

By transforming an image from one space to another, we not only separate different visual properties, but we can also change the predictive (and thus information-containing) properties of the data. DCT transforms the data such that there is more coding or inter-pixel redundancy. By removing these redundancies we can compress the image better.

AC-coefficients close to top left of the frequency spectrum represent highly correlated pixel values (low frequency), while AC-coefficients farther towards the bottom right represent rapidly changing pixel values (high frequency), such as edges and noise. Using DCT, most of the energy is collected in the coefficients near DC, with decreasing energy levels towards the lower right AC coefficient. That is the upper left coefficients are more important to visual quality when restoring the image. Quantization is done by dividing and truncating each of the transformed coefficients by individual values that are given in a quantization matrix, which becomes a part of the compressed stream. Quantization is the greatest source of loss in information, as decimal digits are truncated. The quantization matrix typically contains higher values towards the lower right making several of the less important coefficients with smaller energies equivalent to zero.

3.3. Hadamard Transform

The Hadamard Transform consists of a projection onto a set of orthogonal square waves called Walsh functions and hence the transform is also known as Hadamard transform. The Hadamard transform coefficients are called sequence components, and the Walsh functions are ordered by the number of times they cross zero. The Walsh functions are real and take either the values of +1 or -1. A Hadamard matrix is a symmetric N × N matrix with elements +1 and -1. The Hadamard matrix of order two is given in eqn. (3.3)

$$H_{2,2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.3}$$

$$H_{k,k}^{-1} = \frac{1}{k} H_{k,k} \tag{3.4}$$

The major advantage of Walsh Hadamard transform over other transforms is that the computation required is lesser and these functions have matrix representation which makes things easier to implement in digital computing. This transform requires lesser computation as the function takes +1 or −1 which when implemented in matrix would deal with adders avoiding multipliers not only making it faster but also easier to design and implement the hardware. Hadamard is very useful when used with smaller blocks. The matrix required to calculate the inverse of Walsh Hadamard transform can be easily obtained by using eqn. (3.4), which is normalized Hadamard transform.

3.4. Discrete Wavelet Transform

The wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. Wavelet based schemes, also known as sub-band coding, perform much better when compared to image compression schemes using transforms like DCT. Since the input images are not divided into smaller blocks, coding schemes at higher compression do not produce blocking artifacts, which are clearly visible at high compression rates when using DCT based scheme. Wavelet based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images.

The fundamental concept behind sub-band Coding is to split up the frequency band of an image and then to code each sub-band using a coder with bit rate accurately matches to the statistics of the band. Sub-band coding is used extensively because of its inherent advantages resulting from variable bit assignment among the sub-bands as well as coding error confinement within the sub-bands. Encoding the complete image without separating the image into smaller blocks not only avoids the blocking artifacts but also helps in retrieving the image in different sizes and resolutions from the same compressed data. Some of the basic image processing applications like cropping and rotation of the image can also be easily implemented in the compressed domain, as image is not split into smaller blocks. Fourier transform gives all the frequency components in the signal but fail to represent the occurrence of frequency with respect to time. Wavelet transforms, on the other hand, give a reasonable approximation of the occurrence of the frequency component along with the time of occurrence by analyzing the signal using different resolutions.

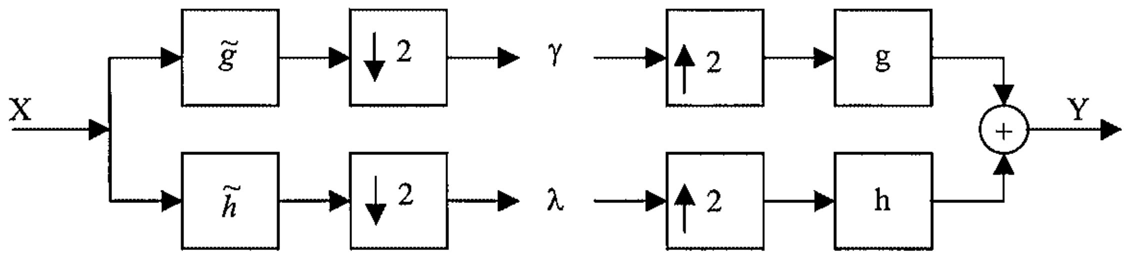


Fig. 3.1: Multi-stage filter bank for signal analysis and reconstruction.

Fig. 3.1 shows the filter banks, which are used to filter the lowest frequency components recursively and down sample all the four outputs. The input image represented by X is passed through high-pass and low-pass filters for rows and columns sequentially resulting in four matrices as outputs. The process of filtering requires convolution of the input matrix with the filter coefficients, which needs lot of computations. To reduce the calculations, lifting coefficients are used. As multiplication with the lifting coefficients is same as the convolution with filter coefficients. In lossless image compression care should be taken to make sure that the lifting coefficients or filtering coefficients do not introduce any rounding errors. The lifting coefficients not only reduce the computation but it reduces the memory required for computation as the coefficients obtained by one level can replace the coefficients obtained by previous level.

These coefficients after recursively being filtered are encoded such that the information can be retrieved in different resolutions and sizes depending on the target device. Encoding also supports progressive transmission, which means the entire image, is displayed with least number of bits with the details added to it progressively with the receipt of further information instead of getting image sequentially from top to bottom. Progressive transmission is also supported in compression techniques implementing DCT but implementation becomes lot easier with wavelets. This mode not only helps in better viewing the data even before the complete data is extracted but also helps in transmitting the image with different sizes and quality depending on the target device from the same encoded data stream. At the decoder side the filters are up-sampled and filtered with the prediction and error merged after filtering to obtain the reconstructed image.

Haar is basic form of wavelet, which is used in image compression with many other wavelet transforms like Welsh, Daubechies that have different set of filters, which satisfy the corresponding wavelet functions. JPEG 2000 uses Daubechies (9,7) wavelet for lossy compression. Daubechies function although represents the image signal well introduces rounding errors in digital computing hence wavelet (5,3) is used for lossless case as it eliminates the occurrence of rounding by using computation requiring integers only. JPEG 2000 also supports customizable wavelet transforms. This means the wavelet to be used for compression can be selected by the user, which need not be a standard wavelet in JPEG 2000 standard.

CHAPTER IV

IMAGE COMPRESSION USING SELF-ORGANIZING NEURAL NETWORK

In lossy compression methods DCT and DWT are widely used to compress images. They use correlation between neighboring pixels to reduce the redundancy in the image and remove the high frequency components that are irrelevant. These transforms do not use the redundancy between different parts of the image that may be similar. There are other techniques like vector quantization that consider this type of redundancy but fail to remove the irrelevant high frequency information from the image. In these techniques the image is first divided into small blocks of size 8×8 and a code book is formed which should be as small as possible but can give a reasonable approximation of the image. Each block in the image can be replaced by its nearest code word at the decoder side. The compressed image contains the code table and index of which code in the code table is nearest to the current block. The flowchart of the implementation is shown in fig. 4.1.

4.1. Proposed Technique

The block diagram for the proposed algorithm is shown in fig. 4.2. The image is first divided into smaller blocks of 8×8 and DCT is applied on these blocks individually. These blocks that contain the DCT coefficients are arranged in an ascending order with respect to their frequencies forming a one-dimensional vector of length 64. These vectors are sent to the classifier, where the blocks are classified to form a code table with the smallest number of codes that can give a reasonable approximation of the input vectors. The codebook and the index that gives the vector code word in the code table that is similar to its corresponding position are sent to the encoder where the data is encoded using first order predictor to attain better compression. At the decoder side, same steps are reversed with the decoder decoding the input to get back the codebook and the index. The declassifier depending on the index table substitutes the code words in the code table. Then the obtained matrix that is of the same size as the image is divided into 8 × 8 blocks and inverse DCT is calculated to get the reconstructed image.

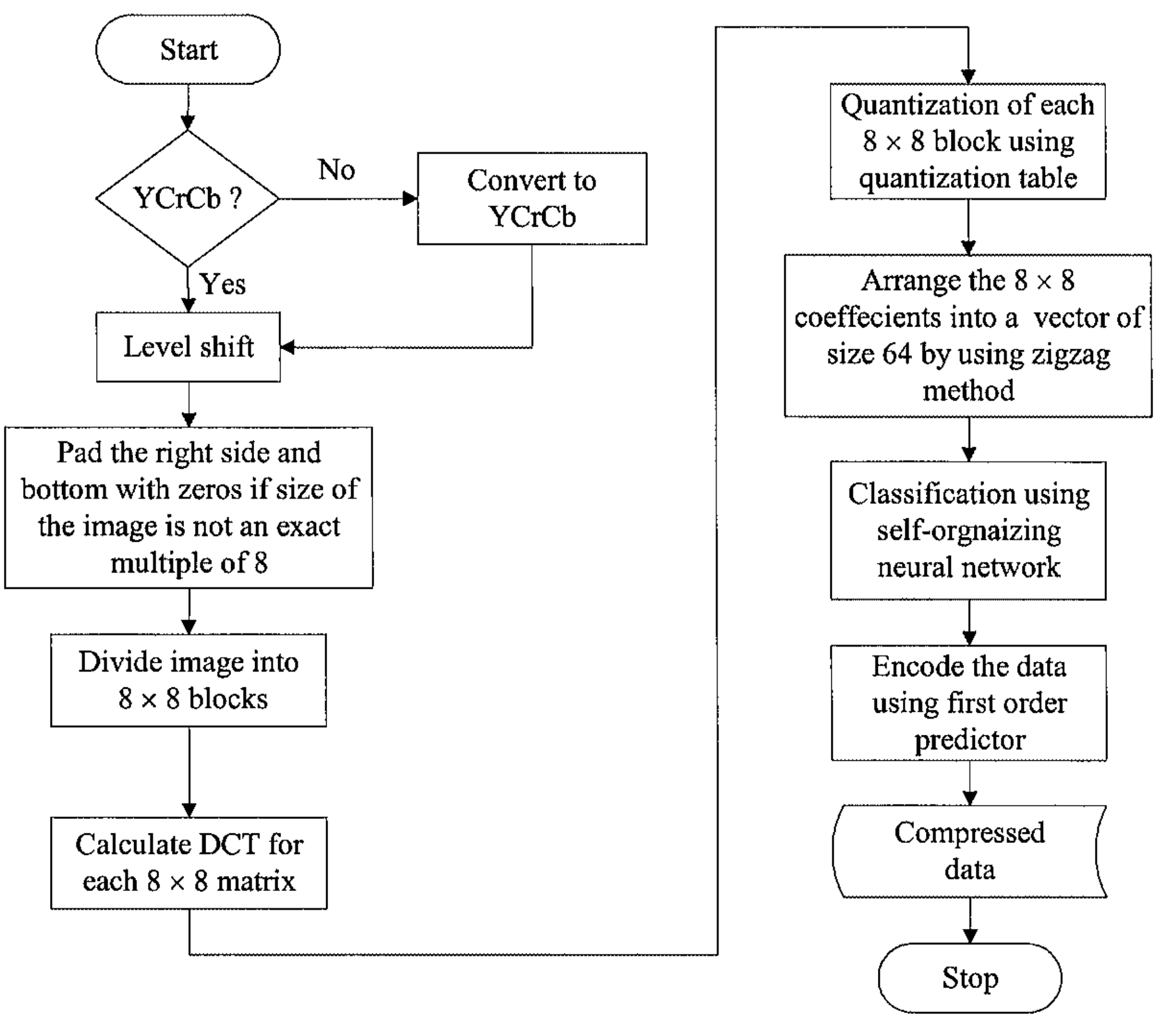


Fig. 4.1: Flow diagram of the encoder.

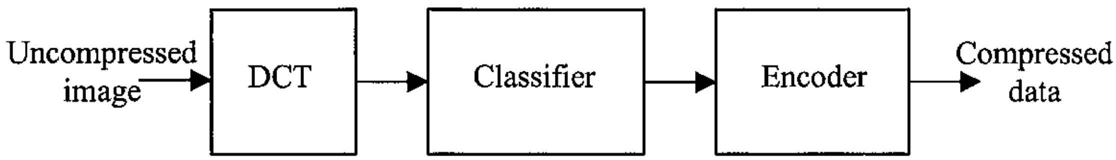


Fig. 4.2: Block schematic representation of the DCT-neural network encoder.

4.1.1. Image Preprocessing

The response of the human visual system is highly dependent on the frequency. Mathematical transforms are applied to signals to obtain the information from the signal that is not readily available in the raw signal. The image when, represented in the frequency domain, is easier to separate into the high frequency (irrelevant data) and low frequency components. DCT transform is suitable for image compression algorithms because it has good energy compaction. As there is low correlation between the coefficients, each coefficient can be treated separately. A small error induced in a coefficient during compression does not have much of an impact on the overall image, as the error from one coefficient is not propagated through out the image.

The algorithm calculates DCT separately for every 8×8 block. The output of the DCT transform is a matrix of size 8×8. The size of the block is a tradeoff between computational speed and compression efficiency. The size of the block can be reduced to 4×4 which will decrease the computation but in turn reduces the compression efficiency. After the DCT is calculated, the horizontal frequency of the basic function increases from top to bottom and the vertical frequencies increase from left to right. So the top left coefficient and the bottom right coefficients represent the lowest and highest frequencies respectively.

The output obtained from the DCT transform gives the coefficients representing the weights of different frequency components in the output signal. These output coefficients are then quantized using standard JPEG quantization matrices. The quantization matrix has large quantization steps for high frequency components and smaller steps for low frequency components, as the human eye is sensitive to low frequency components. The matrix coefficients after quantization are rounded to their nearest integers, resulting in loss of information. The coefficients in the 8×8 matrix are then arranged approximately in ascending order i.e. lowest to highest frequencies by using zigzag method. The transform is lossless as it is reversible but introduces small errors due to computational rounding in digital computers. Information is lost in this block in the quantization phase where coefficients are rounded to nearest integers after dividing the coefficients with quantization matrix.

4.1.2. Self Organizing Neural Network Based Classifier

The function of this block is to find all the blocks in the image, which are similar and cluster them into groups, which is done using neural networks [43 – 47]. Vector quantization and neural networks based on Fuzzy Adaptive Resonance Theory (Fuzzy ART) are used to find the similar blocks in the image [48,49]. As the vectors are the DCT coefficients which are quantized the higher frequency components for most of the vectors will be zero which helps in image compression as only a smaller number of coefficients are required to be similar to represent them as a single code. For higher rates of compression the DCT coefficients representing high frequency components can be conveniently ignored without much loss in visual quality of the image.

Vector quantization is done such that it uses the relation between frequency components and human visual system to optimize the image quality for a given rate of compression. The weight coefficients for lower frequency components are set to be higher than the weight coefficients of the high frequency components. So two blocks that have larger distance in higher frequency components can be grouped as similar as higher frequencies mostly contain irrelevant data but the difference in lower frequencies should be small for them to be considered as similar.

Vector Quantization is a lossy compression method, which classifies data depending on their similarity so that all similar blocks are clustered to reduce redundancy in an image. The neural network based on ART is used to detect the blocks of image that are similar to each other. This neural network depending on the weights of the coefficients finds the similar vectors by using choice functions (CF_j) and vigilance function (Vig_j), formulae are given in eqn. (4.1) and eqn. (4.2) respectively.

$$CF_j = \frac{\sum_{i=1}^{64} |X_i . W_{ij}|}{\sum_{i=1}^{64} |W_{ij}|} \tag{4.1}$$

$$Vig_j = \frac{\sum_{i=1}^{64} |X_i . W_{ij}|}{\sum_{i=1}^{64} |X_{ij}|} \tag{4.2}$$

The algorithm starts by comparing each input vector also called as a node in neural network, with the vectors already in the codebook by calculating vigilance function for each of them. The value of ρ typically varies from 0.8 to 1.0 depending on the compression required. The compression increases with the decrease in the value of ρ, which in turn reduces the quality of the image. With the value of ρ decreasing vectors, which are not relatively similar, are also grouped and represented by single code in the codebook. When a matching node is found with vigilance below ρ, that node is called a winner and it is assumed that current node and the winner node are similar and can be stored as a single block in the codebook instead of two. If more than one code are within the range of ρ the code with the maximum choice function is considered to be a winner. That is the node with maximum choice function is the nearest approximate of the current vector and hence is considered to be a winner. The node that is saved in the codebook is modified to be the mean of all the vectors previously grouped in that block and the new block calculated by the expression shown in eqn. (4.3). Hence the value in the code table is a good approximation of not only the new vector in the code table but also the n vectors already found to be similar to the particular vector in the codebook. Fig. 4.3 shows the schematic diagram of the Fuzzy ART classifier.

Fuzzy logic is used in this classifier to make the system computationally efficient by using analog "and" to reduce the multiplications required for finding the minimum value of the two functions, which give a reasonable approximation with much lesser computation.

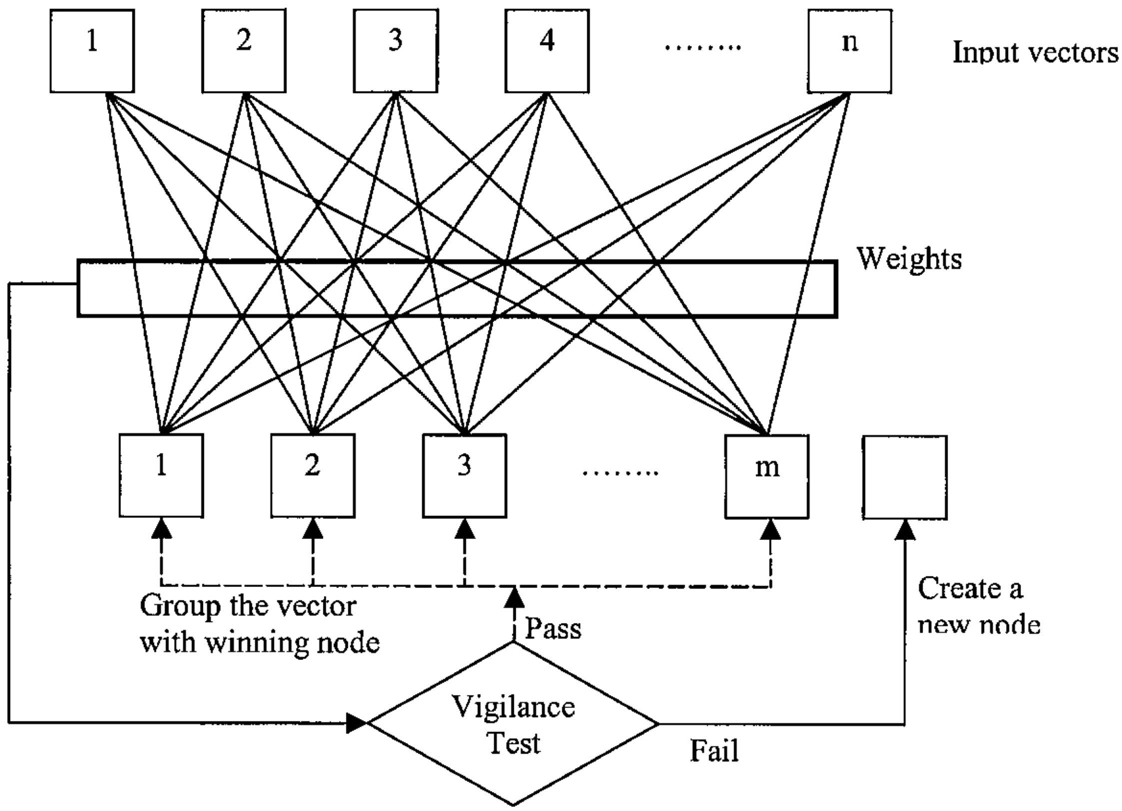


Fig. 4.3: Schematic representation of the neural network classifier.

When a new node is found to be similar to an existing node the weight of the winner node is modified using eqn. (4.3).

$$W_{ij}^{New} = \frac{nW_{ij}^{old} + X_{ij}}{n+1} \tag{4.3}$$

If there is no match found after completing all the blocks, then that block is not similar to any of the existing nodes in the codebook. So a new entry with this vector is added to the code table. This process is repeated until all the blocks are compared and their node numbers are stored as an index that gives the position of the vector in code table that is nearest to the corresponding block. This process is repeated iteratively a few times to train the neural network.

The neural network based on Fuzzy ART is self-organized, which means the neural network can train itself based on the input vectors. The incorporation of a vigilance function acts as a tolerance measure to make the neural network to accept new information available without destabilizing the neural network, which would result in loss of all prior information. The output of this block is the matrix, which contains the index and the codebook. The index refers to the code in the codebook to which the block in the corresponding location is similar. This block is the lossy part of the image as it considers vectors that are different to be same as they are nearer to each other, losing the data that is different between the two blocks.

4.1.3. Coder based on First Order Predictor

The purpose of this block is to code the code table and the index of all the blocks to compress them efficiently to attain even higher compression [50,51]. As in most natural images, the gradient of the image changes smoothly and hence the change in gradients between adjacent blocks should be small and constant. The ordinary run-length encoding which uses zero order prediction does not effectively take advantage of this redundancy. The predictor method proposed is of first order, which not only effectively takes advantage of the similarity between any of the adjacent blocks but also works well for cases where the change is gradual. The information in the first two lines is required for getting first order prediction for the third line. First two lines of the index table are coded using normal run-length coding with the sequence taken to be zigzag as shown in the image. This zigzag pattern is found to increase the compression as it tends to group similar blocks continuously when compared to conventional run-length coding.

Traditional run-length sequence coding sequence and the modified run-length coding sequence are shown in fig. 4.4 and fig. 4.5 respectively.

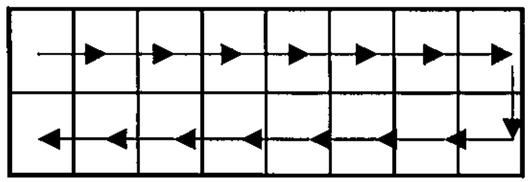


Fig. 4.4: Conventional run-length coding sequence.

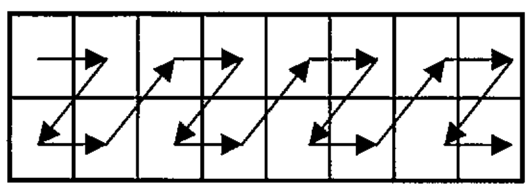


Fig. 4.5: Modified run-length coding sequence.

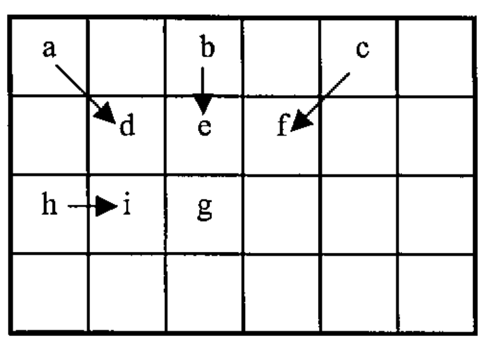


Fig. 4.6: Determination of the best direction to calculate the difference.

After the first two lines are encoded from the third line first-degree predictor is used instead of zero degree predictor. The encoding scheme using first-degree prediction is implemented by finding the change in gradient for each direction. From fig. 4.6 the change in each gradient is calculated by subtracting a-d, b-e, c-f and h-i. The index at the position g is predicted to have the smallest change in gradient in the direction of the smallest change. The error in prediction will be coded at g instead of its original index. The value of g in many cases will be zero as the change in gradient is likely to be similar for adjacent blocks and more over the direction need not be encoded in the compressed bit stream as at the decoder all the blocks required are already decoded. The block diagram for calculating the code used to encode the index table is shown in fig. 4.7.

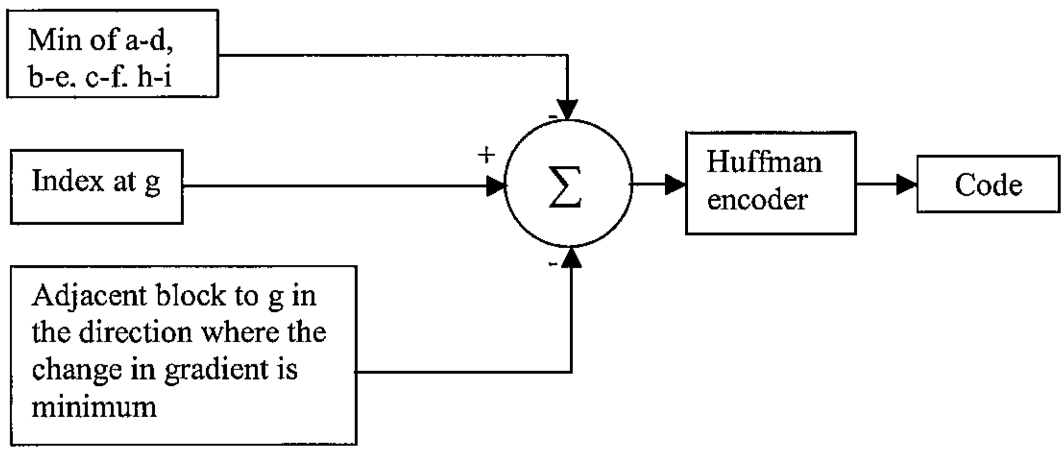


Fig. 4.7: Coding of the difference between blocks.

Then Huffman encoder, which uses variable length coding where the length of the code is inversely proportional to the probability of the occurrence of a symbol, is used to encode this coded data. The code table is coded using symbol encoding to attain higher compression with the lowest frequency component also called the DC coefficients of all the codes coded separately as their magnitude is much higher compared to other coefficients. At the decoder, image is reconstructed by first decoding the Huffman coding and then using the inverse predictor method to get back the indices. Then the image can be reconstructed by placing the nodes corresponding to the index in that particular block and multiplying the blocks with quantization matrix and calculating the inverse DCT for each block. This process of encoding is lossless.

This compression technique is not symmetric as it requires a lot of computation when encoding but the computation at the decoder side is minimal as it just has to replace the block with the corresponding code word from the code book. The flowchart for the decompressor is shown in fig. 4.8 and the block schematic diagram of the decoder is shown in fig. 4.9.

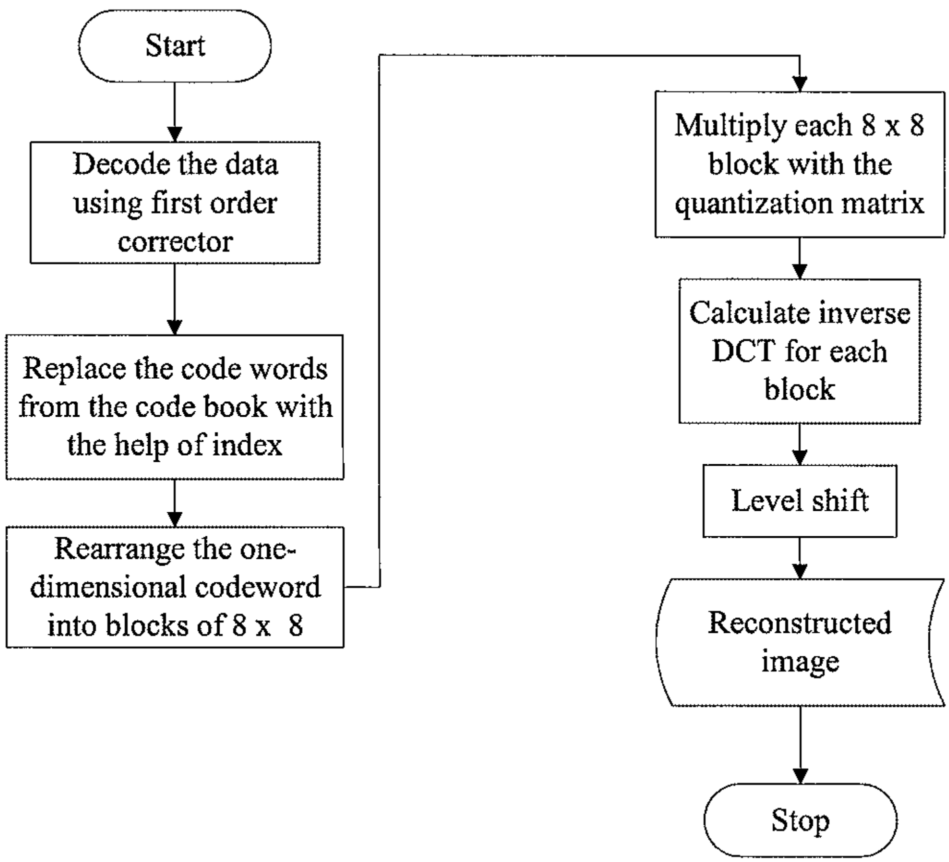


Fig. 4.8: Flow diagram of the decoder.

4.1.4. Decoder for the First Order Predictor

The purpose of the decoder is to decode the variable length Huffman code to get back the predictor code. The first two lines of the index table can be obtained by decoding the run length code and then arranging the indices using the zigzag method at the encoder. Fig. 4.10 shows the block diagram of the algorithm to get the index from the error code.

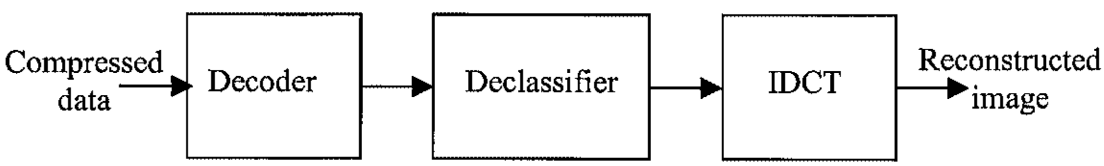


Fig. 4.9: Block schematic representation of the DCT-neural network decoder.

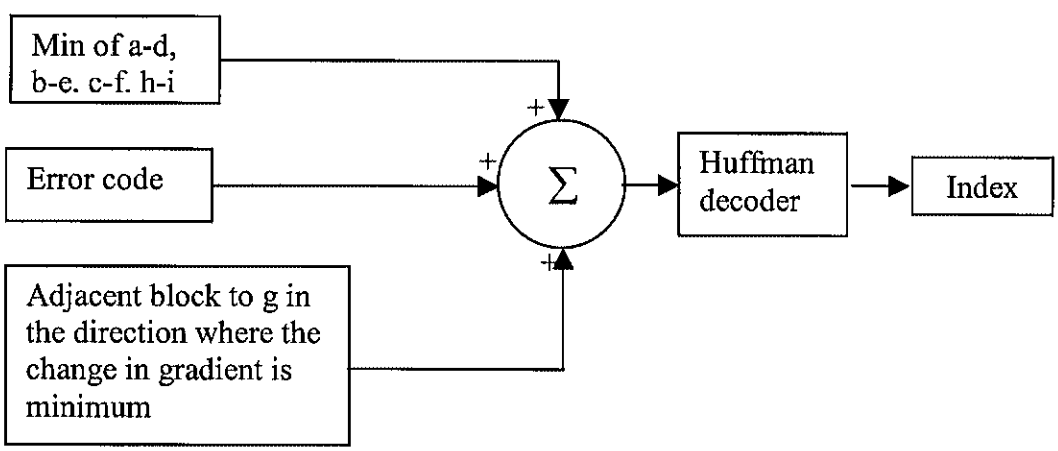


Fig. 4.10: Calculating the best direction with minimum distance.

At the decoder side as the information like the index above and to the left of the current index are already determined, the index of the current block can be obtained as shown in fig. 4.10. The change in gradient for all the directions is found out by calculating a-d, b-e, c-f and h-i as in the coder from which the direction with least gradient can be obtained. The predictor at the encoder predicted the value of current block to hold good for least gradient and the error in the prediction is coded as the error code. The index can be obtained by adding the error code with the least change in gradient and the index of the block adjacent to current block in the direction of least gradient. As in most cases, the change in gradient is smooth and the direction of smallest change in gradient is likely to occur. The error produced using this predictor method has small values except in cases where there is a sudden change in the gradient. Most of the images used are natural images and are more likely to have a smooth change in gradient unlike artificial images where the gradient change is abrupt and sharp. The compressed data is decoded here such that it has an index, which represents the block in the codebook to which the corresponding block in original image is similar along with the codebook, which contains all the nodes required to reconstruct the image. The DC coefficients, which are encoded differently from the other coefficients, are decoded separately and merged together. The output of this block is exactly the same as the input to the encoder as the encoder does not introduce any error while encoding the data.

4.1.5. Declassifier

The input of this block is an index table, which represents the block in the codebook. The purpose of this block is to replace the index with the code in the codebook represented by the index. This block makes this compression technique dissymmetric, as the time required to classify similar blocks requires lot of computation at the encoder. At the decoder, only the blocks have to be fetched from the codebook and reconstructed with the help of the index table. The output of this block is similar to the input, which was given to the classifier at the time of compressing the data with the error introduced at the classifier when two blocks, which are similar but not the same, are represented using the same code.

4.1.6. Image Postprocessing

This block reconstructs the image using DCT coefficients obtained from the declassifier. At the encoder side if the coefficients with higher frequency are ignored to attain higher compression ratio, these higher frequency components are replaced with zeros. These coefficients are then arranged into 8×8 matrix by using reverse zigzag process. The 8×8 matrix obtained is multiplied with quantization steps from the quantization matrix, which was used at the DCT block. The loss of information due to the rounding at the encoder side after quantization results in loss of information, which cannot be recovered. The inverse DCT is calculated from these coefficients by using the formula in eqn. (2.2). After the inverse DCT is calculated the output of the IDCT block is the reconstructed image. The image obtained will have block artifacts, which are dominant if the compression required is high. The filtering of the image is performed to smoothen the image, which may result in further loss of some more information, but this process makes the distortions in the image less obvious to the human eye by smoothening the abrupt changes.

CHAPTER V

EXPERIMENTAL RESULTS

5.1 Measure of Image Quality

Signal-to-noise (SNR) measure gives the estimate of the quality of a reconstructed image compared with an original image. The basic idea is to compute a number, which can reflect the quality of the reconstructed image. Traditional SNR measures do not equate with human subjective perception. Hence the higher measures do not always mean better quality as it is just an approximation of the quality of an image but not an accurate representation. The actual metric computed is the peak signal-to-reconstructed image measure, which is called PSNR. For the source image f(i,j) that contains N by N pixels and a reconstructed image F(i,j) where F is reconstructed by decoding the encoded version of f(i,j). PSNR is computed by computing the mean squared error (MSE) of the reconstructed image using eqn (5.1). The root mean square error (RMSE) is the square root of the MSE. PSNR in decibels (dB) is computed using eqn (5.2).

$$MSE = \frac{\sum[f(i,j)-F(i,j)]^2}{N^2} \tag{5.1}$$

$$PSNR = 20\log_{10}\left(\frac{255}{RMSE}\right) \tag{5.2}$$

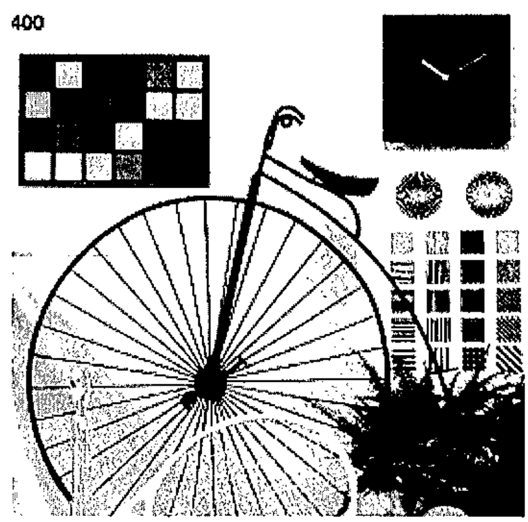
The actual PSNR value gives a rough estimate of the quality of the image, but the major advantage of it is in comparison between two values for different reconstructed images as it gives a comparative measure of quality. Some definitions of PSNR use 255^2/MSE instead of 255/RMSE. Both of these formulations work as the concentration is on comparing PSNR values of two reconstructed images. The other important technique for displaying errors is to construct an error image, which shows the pixel-by-pixel errors in the image. This can be done by taking the difference between the pixels of the reconstructed and the original image. This image that represents the error for each pixel is hard to see as zero difference is represented by black and most errors are small numbers, which are shades of black and cannot be easily distinguished. Hence the error image multiplies the difference by a constant to increase the contrast and add a constant to this image to convert signed integers in the error matrix to unsigned integers, which make the image comprehensive. The error matrix is obtained using the eqn in 5.3.

$$E(i,j) = 2\,[\,f(i,j) - F(i,j)\,] + 128 \tag{5.3}$$

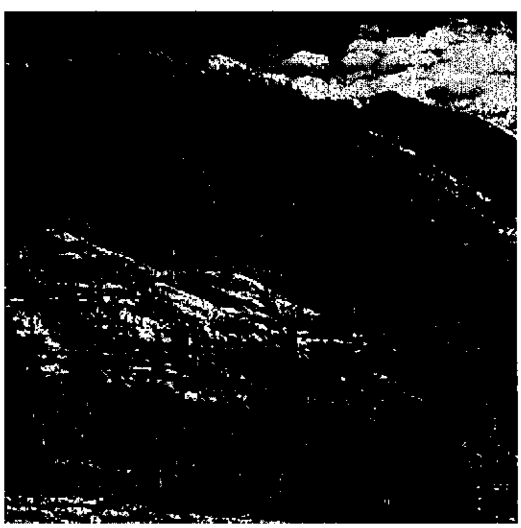
5.2 Comparison of Image Compression using Different Transforms

The performance of transform-based techniques using Hadamard, Hartley, DCT and Haar transforms for 10 different test images is tabulated in table 5.1. The results show that the compression technique based on Haar wavelet performs better when compared to other transforms. Haar wavelet performs better for most of the images due to the properties of wavelet, which represent the signal in different resolutions helping it in better representing the image even in compressed domain. The table clearly demonstrates that image compression scheme using Haar outperforms other transform-based techniques as reconstructed images with better quality are attained for the same rate of compression. All the comparisons made in this chapter use the ten test images given in fig. 5.1.

Test image 1.



Test image 2

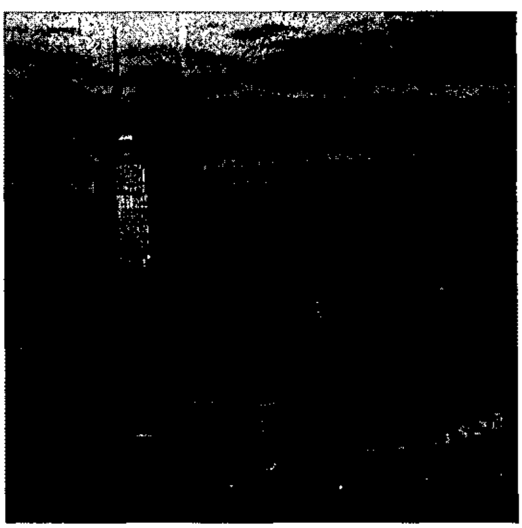


Test image 3.

Test image 4.

Test image 5.

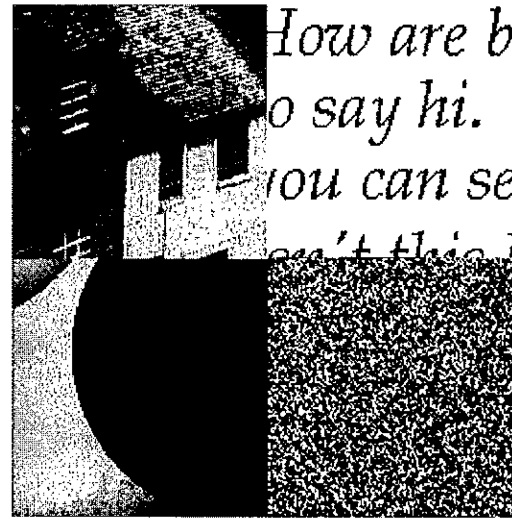
Test image 6.



Test image 7.

Test image 8.

Test image 9.



Test image 10.

Fig. 5.1: Test images for performance comparison.

The DCT based transform produces reasonably good quality at low and medium rates of compression but suffers from blocking artifacts at higher rates of compression where Haar has a definite advantage due to bigger tile sizes. DCT transform does a good job and is better than Haar on images with energy mainly concentrating on low frequency components. At higher rates of compression, the DCT transform produce blocking artifacts, which can be smoothened by low pass filtering the reconstructed image. This process will induce an error into the reconstructed image but the resulting image is more pleasing to human eye as it reduces the abrupt changes near the edges of the blocks.

In Hartley transform based compression technique, the computation required is high as it has to compute both cosine and sine components instead of just the cosine transform as in the case of DCT. The quality of the reconstructed image is reasonable but lesser than the compression techniques based on DCT and Haar. This transform tends to perform better with images where change of gradient is smooth.

The major advantage of Hadamard over other types of compression algorithms is that the computation required is minimum as it takes values of 1 or −1, which means only computations required are additions and subtractions. The compression attained by this process is lesser compared to other transform based compression techniques. Hadamard based compression can be used when fast encoding and decoding of the image is required to minimize the hardware requirement.

Table 5.1 PSNR for 10 test images when compressed at 1 bpp.

Test image	Hadamard	Hartley	DCT	Haar
1	30.24	31.45	33.03	32.24
2	25.86	26.13	27.55	30.60
3	24.52	24.08	25.23	25.98
4	24.47	25.17	25.81	26.58
5	24.38	24.86	26.18	27.27
6	37.13	38.52	38.73	40.217
7	32.70	34.21	35.25	36.921
8	19.86	19.018	20.53	20.989
9	24.89	25.62	26.45	27.486
10	22.96	24.68	27.24	28.12

5.3 Results for the Proposed Technique

The results obtained by the proposed technique at various compression rates are given in tables 5.2 to 5.4 and are compared with standard image compression algorithms like JPEG and JPEG 2000. The table shows image quality of the reconstructed image when compressed to 1, 0.5 and 0.25 bpp using different image compression techniques.

Table 5.2 PSNR for 10 test images when compressed at 1 bpp.

Test image	JPEG (DCT)	JPEG 2000 (5/3 wavelet)	JPEG 2000 (9/7) wavelet	(DCT+NN)
1	33.03	34.12	38.16	36.38
2	27.55	30.44	30.65	29.73
3	25.23	26.31	27.09	25.76
4	25.81	26.67	26.98	26.17
5	26.18	27.23	27.65	26.68
6	38.73	40.71	40.87	39.56
7	35.25	36.74	37.52	36.32
8	20.53	20.99	21.93	21.79
9	26.45	28.20	28.66	28.09
10	27.24	27.65	28.79	**29.17**

Table 5.3 PSNR for 10 test images when compressed at 0.5 bpp.

Test image	JPEG (DCT)	JPEG 2000 (5/3 wavelet)	JPEG 2000 (9/7) wavelet	(DCT+NN)
1	28.25	29.41	30.61	29.15
2	23.40	25.52	25.63	24.86
3	22.54	22.99	23.45	23.14
4	24.13	24.01	24.60	**24.73**
5	22.69	23.21	23.80	23.26
6	36.15	37.49	37.45	36.47
7	32.81	33.44	33.73	33.52
8	19.62	18.99	19.03	**20.87**
9	24.51	24.57	24.97	**25.16**
10	22.54	21.42	22.37	**23.27**

Table 5.4 PSNR for 10 test images when compressed at 0.25 bpp.

Test image	JPEG (DCT)	JPEG 2000 (5/3 wavelet)	JPEG 2000 (9/7) wavelet	(DCT+NN)
1	25.08	25.73	27.13	25.67
2	21.14	21.532	21.80	21.34
3	21.13	21.02	21.44	21.74
4	22.51	22.49	22.8	**23.08**
5	20.88	21.18	21.17	**21.23**
6	33.51	34.48	35.05	33.87
7	30.42	30.67	31.03	30.71
8	17.52	17.43	17.92	**18.14**
9	22.05	22.28	22.71	**22.97**
10	19.24	18.385	18.82	**20.07**

(a) Original image (b) Reconstructed image (c) Error image

Fig 5.2: Image compressed using conventional JPEG.

(a) Original image (b) Reconstructed image (c) Error image

Fig 5.3: Image compressed using 5/3 wavelets.

(a) Original image (b) Reconstructed image (c) Error image

Fig 5.4: Image compressed using 9/7 wavelets.

(a) Original image (b) Reconstructed image (c) Error image

Fig 5.5: Image compressed using a combination of DCT and neural network.

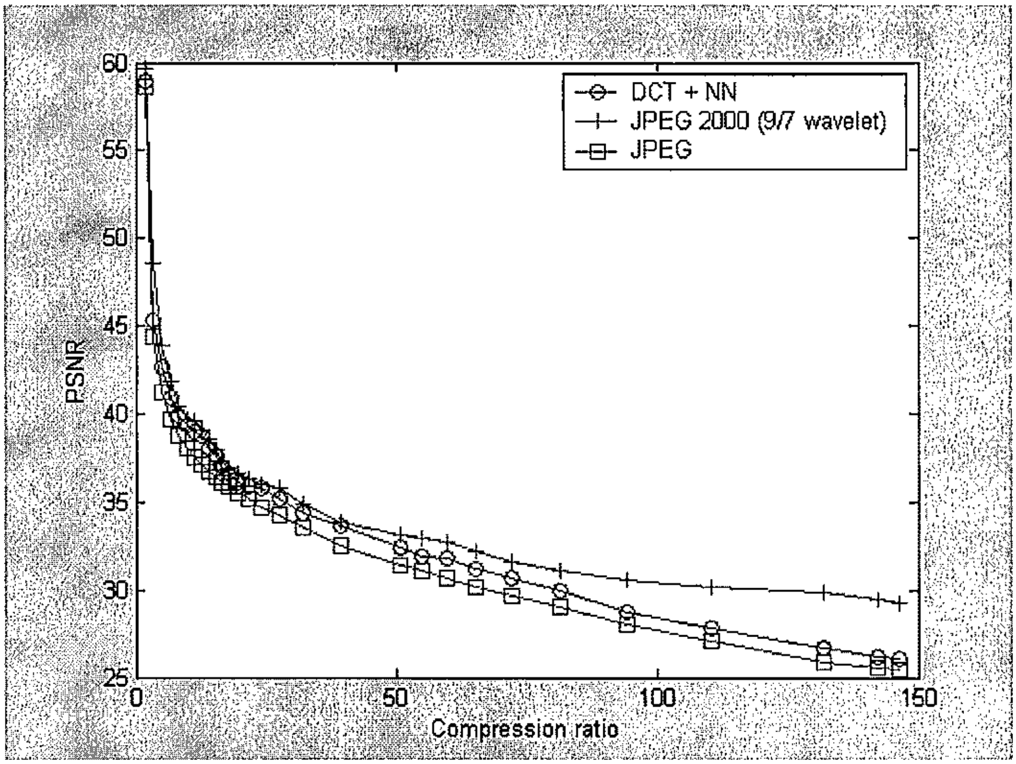


Fig. 5.6: Effect of compression ratio on PSNR for test image 6.

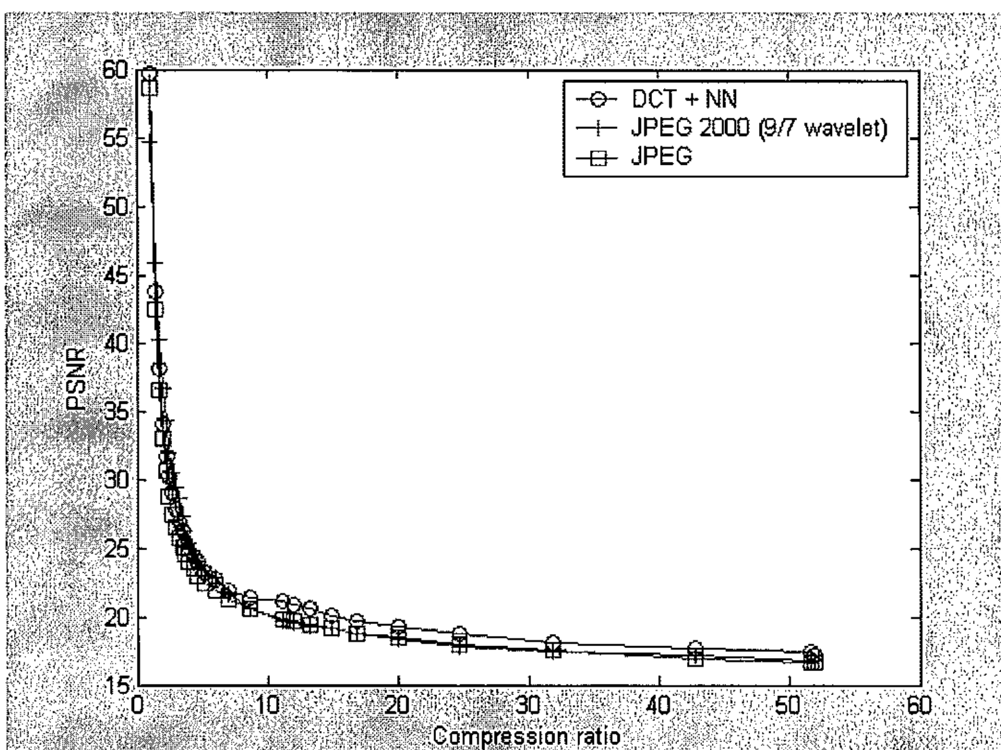


Fig. 5.7: Effect of compression ratio on PSNR for test image 8.

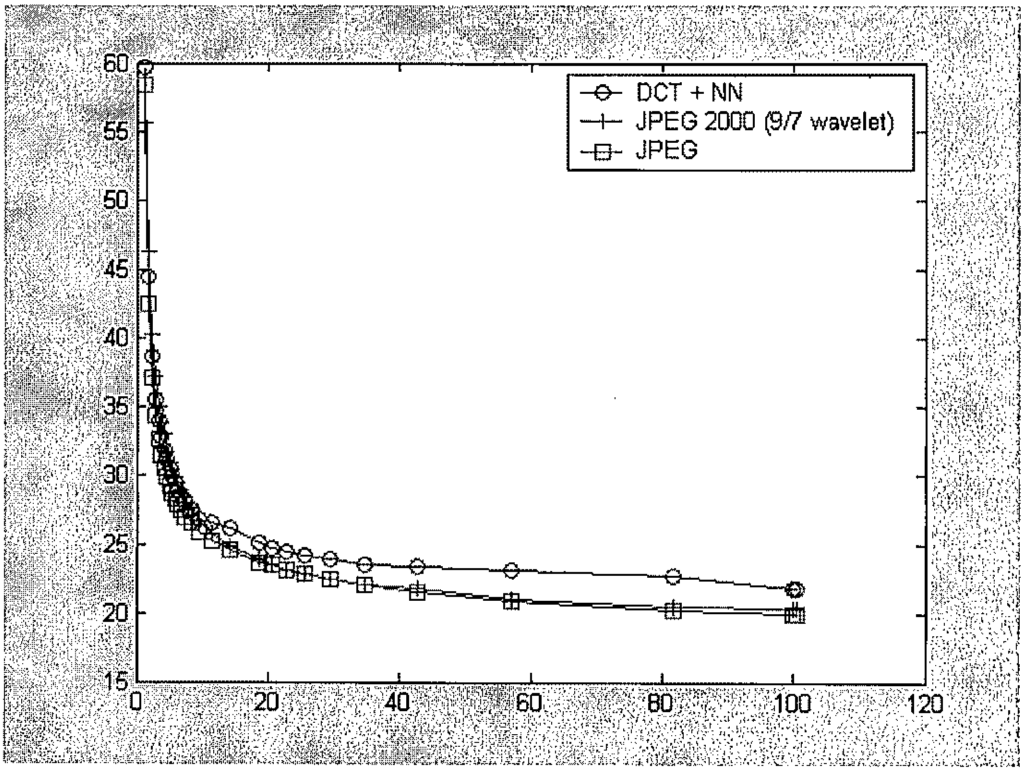


Fig. 5.8: Effect of compression ratio on PSNR for test image 9.

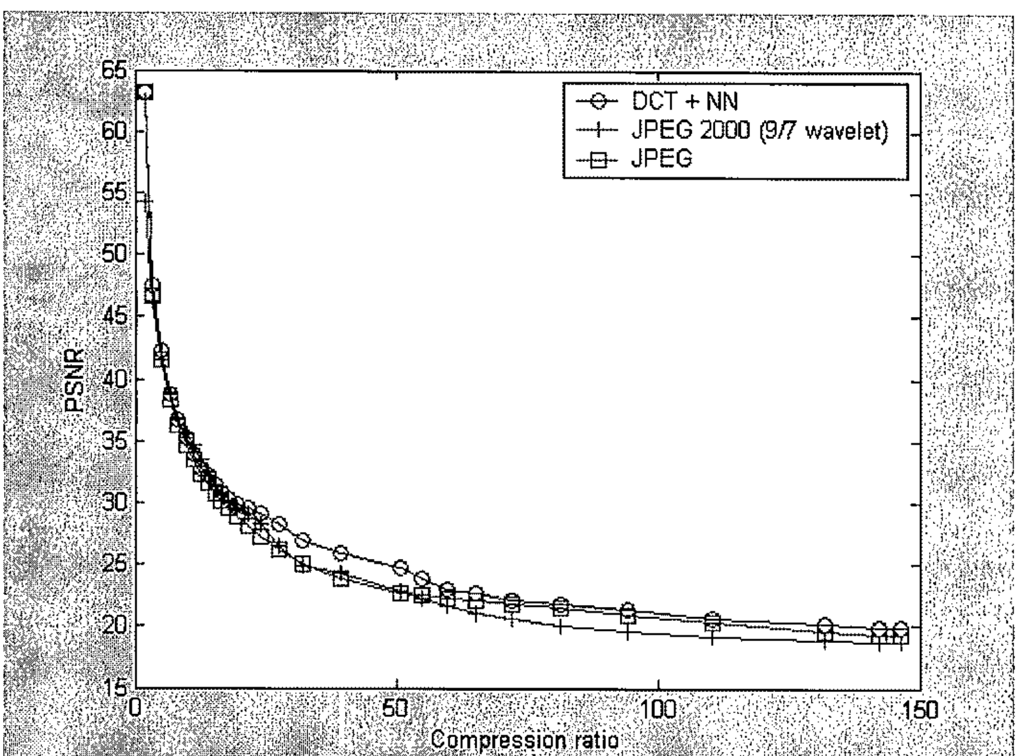


Fig. 5.9: Effect of compression ratio on PSNR for test image 10.

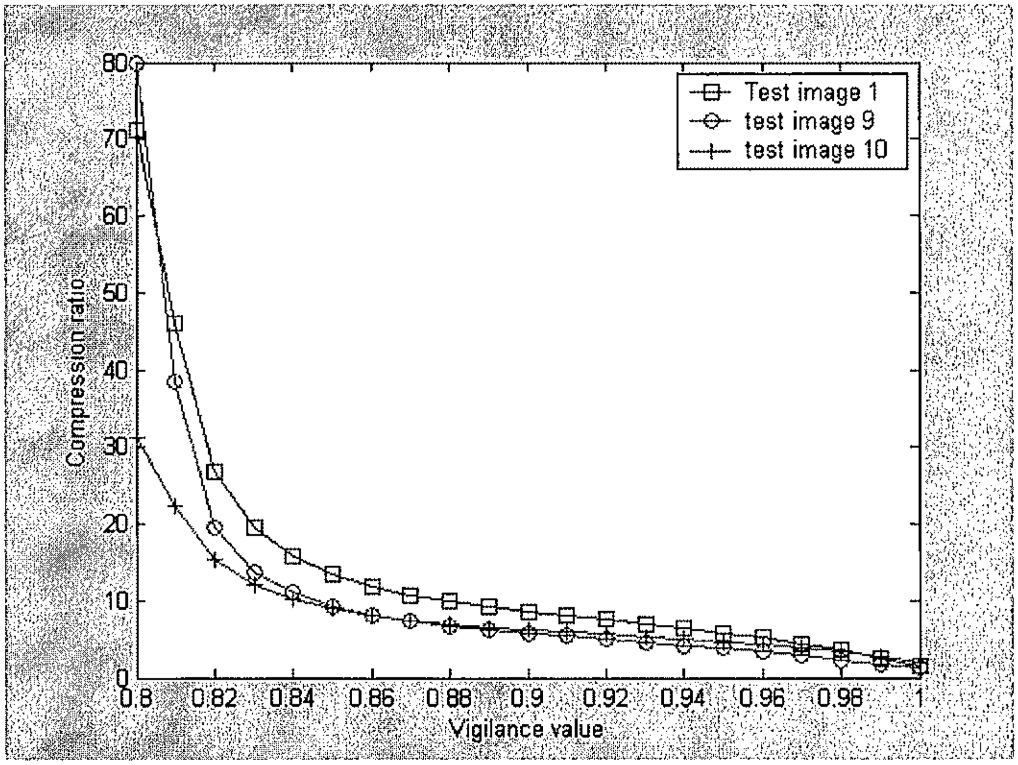


Fig. 5.10: Effect of vigilance value on compression ratio.

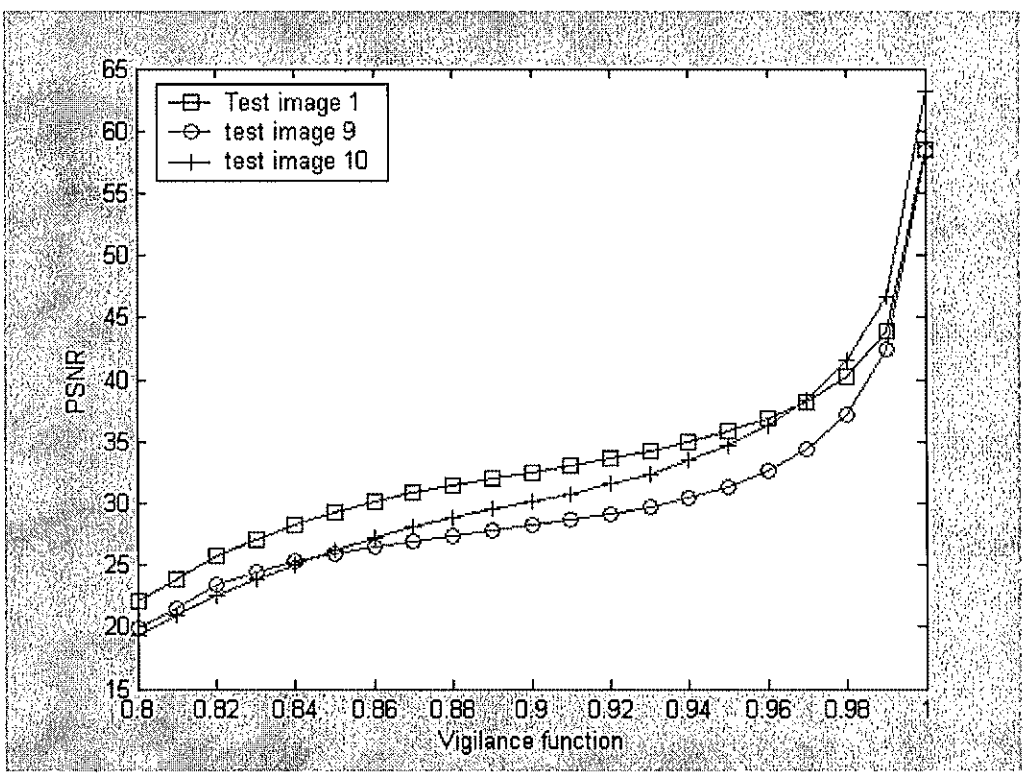


Fig. 5.11: Effect of vigilance value on PSNR.

The quality of the reconstructed image using the proposed research is better compared to the conventional DCT based JPEG compression algorithm and is comparable to the wavelet based JPEG 2000 standard. Figures 5.2 to 5.5 show the original image, reconstructed image and the error in the reconstructed image respectively for different standard image compression techniques. Fig. 5.6 to 5.9 show the effect of compression using different techniques on PSNR for 4 different test images. Fig. 5.10 shows the variation in the compression ratios with change in vigilance value. Fig. 5.11 shows the variation in PSNR with change in vigilance value.

At lower rates of compression, JPEG 2000 has an advantage over the proposed algorithm, as the number of code words in the codebook is high. At lower rates of compression, the codebook is big and as the index has to represent many code words coding is not effective. Hence this algorithm performs poorly at lower rates of compression. The time required at low rates of compression is very high as all the input vectors have to be compared with all the code words in the codebook and number of code words in codebook is high at low rates of compression

At higher compression rates of compression, the size of the codebook reduces and the index has a lot of values repeating which can be encoded effectively to compress the image effectively. For higher rates of compression the high frequency components are ignored before classifying which not only reduces the size of the codebook but also the time required for computation is reduced with the vector size becoming small. This approach of removing the high frequency components completely to attain high rates of compression has a marginal gain in the quality of the reconstructed image and the time required for encoding the data reduced considerably. At these rates of compression this algorithm performs at par with the wavelet based compression scheme and even better than the JPEG 2000 for images with high frequency components with a number of similar blocks scattered through out the image.

At very high rates of compression, the time required by this algorithm still reduces as the number of code words in the codebook are even reduced and the values in the index table are repeated more often helping the encoder code them efficiently. But at these high rates of compression, the visual quality of the image reduces considerably. The blocking artifacts at the edges of the blocks become more obvious as this technique splits the image into smaller blocks to reduce the computation required for DCT. Wavelet based technique does not suffer from blocking artifacts as the tile sizes used are much larger compared to the block sizes. These blocking artifacts have a huge impact on human eye as the gradient of the image changes abruptly at the edges of the blocks. The blocking artifacts can be smoothened by passing the reconstructed image through low pass filter to make the transition from one block to another smooth, which is pleasing to human visual system.

The computation time required by this method is high at the encoder side compared to other existing standards. This is not a major drawback, as the computations required for training the neural network with properties of the image are high but the time required at the decoder side is almost same as that required for conventional JPEG based algorithm. As an image is normally encoded once and decoded every time, an image is retrieved from archives or downloaded from the internet the time required for encoding is not as important as the time required for decoding. This compression technique is capable of obtaining results which are much better compared to the conventional JPEG algorithm and is comparable to the JPEG 2000 using discrete wavelet transforms. The method also has a considerable advantage over JPEG 2000 in terms of computation, as it requires fewer computations to decode the image. The DWT based method on the other hand takes almost the same time for decoding as it has taken for encoding. The hardware required by the proposed algorithm is much lesser than the DWT based method. The only thing to be done at the decoder is to replace the block with nearest code word and then calculate the inverse DCT. The information about the nearest code word is stored is found in the index table. The time required for encoding can be reduced if the properties of all the images to be encoded are similar and the encoder has prior information of the characteristics of the image. If all the images with similar characteristics are to be encoded the time required for training the neural network is reduced.

JPEG 2000 has considerable advantage over this method in terms of the number of target devices. Image information can be sent to various target devices as per their requirements from the same compressed JPEG 2000 data stream. JPEG 2000 also supports features like cropping and rotation of an image in compressed domain, which are difficult to implement using the proposed algorithm. Hence this algorithm is not suitable for use in areas where images are to be sent to different target devices. Another major disadvantage of the neural network based method is that it is difficult to control the rate of compression as the compression attained depends on the characteristics of the image and there is no way to control the compressed size or quality of the image.

CHAPTER VI

CONCLUSION AND FUTURE RESEARCH

The main aim of this thesis was to attain better compression by using a combination of transform based compression techniques and vector quantization based techniques. This has been accomplished by developing an algorithm, which uses Fuzzy ART neural network to classify the quantized DCT coefficients to take advantage of both the compression techniques. This yielded better image quality for a given rate of compression.

The algorithm described uses both the properties of vector quantization and DCT to increase the compression by taking advantage of the similar blocks located throughout the image to obtain better compression ratio for a given image quality or a better quality image at a given rate of compression. This algorithm as in all image compression techniques is image dependent and the efficiency increases with the increase in the number of similar blocks even if they are scattered all around the image. The results obtained are better compared to that of the conventional JPEG algorithm. The resultant images are comparable to the wavelet based JPEG 2000 standard, at high rates of compression.

The first order predictor method used to encode the index table takes advantage of the fact that there is a smooth variation in the gradient of the image. As the change in gradient is smooth chances of neighboring blocks being encoded with the same codeword as one of its neighboring blocks is high. This results in better encoding of the index table increasing compression efficiency. As the number of code words are reduced, the chance of neighboring blocks being represented by same code word increases. The first order predictor takes advantage of this to better encode the image.

This neural network based technique would be suitable in applications where a reasonable amount of compression is required with the image having to be decoded many times from the encoded data stream. This algorithm can also be effectively used when compressing group of images with similar characteristics. If the images to be encoded are quite similar, a single codebook can be used for a group of images instead of a codebook for each image.

The proposed algorithm can be put to maximum use by finding the optimum quantization matrix at the quantizer and the tolerance level set at the neural network based classifier. The quality of an image varies a lot with change in these values for the same compression rate. Research work is in progress to improve the performance of the algorithm by incorporating an optimum quantization matrix and an adaptively computed vigilance value. Other transforms like Haar can be implemented instead of DCT to improve the efficiency in compression and also reduce the processing time.

REFERENCES

[1]. R. C. Gonzalez and R. E. Woods, "Digital Image Processing," Englewood Cliffs, NJ: Prentice-Hall, 2003.

[2]. E. R. Dougherty, C. R. Gieardina, "Image processing: continuous to discrete," Englewood Cliffs, NJ, Prentice-Hall, 1987.

[3]. R. J. Clarke, "Digital compression of still images and video," Burlington, MA, Academic Press, 1995.

[4]. S. M. Lei, M. T. Sun, "An entropy coding system for digital HDTV applications," IEEE Transactions on Circuits and Systems in Video Technology, vol. 1, pp. 147-155, 1991.

[5]. B. Macq, "A universal entropy coder for transform on hybrid coding," Proceedings of the Picture Coding Symposium, pp. 12.1.1 - 12.1.2, 1990.

[6]. P. Delogne, B. Macq, "Universal variable length coder for an integrated approach to image coding," Proceedings of the Annals of Telecommunication, pp. 452-459, 1991.

[7]. T. M. Cover, J. A. Thomas "Elements of information theory," John wiley & sons, New York, 1991.

[8]. M. Nelson, J. L. Gailley "The data compression book," M & T Publishing Inc, New York, 1991.

[9]. G. K. Wallace, "The JPEG still picture compression standard," Communications of the ACM, vol.34, pp. 30-44, 1991.

[10]. W. B. Pennebaker, J. L. Mitchell, "JPEG still image data compression standard,"

Van Nostrand Reinhold, New York, 1993.

[11]. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Transactions on Computers, vol. 23, pp. 90-93, 1974.

[12]. K. R. Rao and P.Yip, "Discrete cosine transform algorithms, advantages and applications," Burlington, MA, Academic Press Inc, 1990.

[13]. M. J. Narasimha and A. M. Peterson "On the computation of discrete cosine transform," IEEE Transactions on Communications, vol. 26, pp. 934-936, 1978.

[14]. D. S. Taubman and M. W. Marcellin, "JPEG 2000: Image compression fundamentals, Standards and practice," Kluwer International Series, Norwell, MA, 2001.

[15]. R. A. Devore, B. Jaweth, and B. J. Lucier. "Image compression through wavelet transform coding," IEEE Transactions on Information Theory, vol. 38, pp 719-746, 1992.

[16]. D. Lazr, M. Istaeli "Image compression using wavelet decomposition" IEEE Transactions on Image Processing, vol. 5, pp. 4-15, 1996.

[17]. M. Boliek, C. Christopoulos and E. Majani, "JPEG 2000 Part I final draft international standard," JPEG Committee, 2000.

[18]. A. R. Calderbank, I. Daubechies, W. Sweldens and L. Yeo, "Lossless image compression using integer to integer wavelet transforms," Proceedings of IEEE International Conference on Image Processing, pp. 596-599, 1997.

[19]. W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," Applications of Computing Harmonic Analysis, vol. 3, pp.186-200, 1996.

[20]. W. Sweldens, "The lifting scheme construct or second generation wavelets," SIAM Journal on Mathematical Analysis, vol.29, pp 511-514, 1992.

[21]. R. Claypoole, R. Baraniuk and R. Nowak, "Adaptive wavelet transforms via lifting," Rice University Technical Report # 9304, 1999.

[22]. Allen Gersho and Robert M. Gray, "Vector quantization and signal compression," Kluwer Academic Publishers, Norwell, MA, 1991.

[23]. J. Buhmann and H. Kiihnel, "Complexity optimized vector quantization: A neural network approach," Data Compression Conference, IEEE Computer Society Press, pp 12-21, 1992.

[24]. T. Hofmann and J. M. Buhmann, "Competitive learning algorithms for robust vector quantization," IEEE Transactions on Signal Processing, vol. 46, pp 1665-1675, 1998.

[25]. R. M. Gray, "Vector quantization," IEEE Acoustics, Speech, and Signal Processing Magazine, pp. 9-31, 1984.

[26]. N. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organizing feature maps," IEEE International Conference on Neural Networks, vol. 1, pp. 101-108, 1988.

[27]. N. Farvardin. "A study of vector quantization for noisy channels," IEEE Transactions on Information theory, vol. 36, pp 799-809, 1990.

[28]. P. A. Chou, R. Lookabaugh and R. M. Gray, "Entropy-constrained vector quantization." IEEE transaction on Acoustics, Speech and Signal Processing, vol. 37, pp. 31-42, 1989.

[29]. K. L. Oehler and R. M. Gray, "Combining image classification and image compression using vector quantization," Proceedings of the IEEE Data Compression Conference, pp. 2-11, 1993.

[30]. Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communications, pp. 84-95, 1980.

[31]. J. Jiang, "Image compression with neural networks – A survey," Journal of Signal Processing: Image Communication, vol. 14, pp. 737-760, 1999.

[32]. K. Mehrotra, C. K. Mohan and S. Ranka, "Elements of artificial neural networks," The MIT Press, Cambridge, MA, 1996.

[33]. K. Min., and H. Min, "Neural network based image compression using AMTDAP 610", Application of Artificial Neural Networks, pp. 376-385, 1992.

[34]. J. Jiang, "A neural network design for image compression and indexing", International Conference on Artificial Intelligent Expert Systems and Neural Networks, pp. 296-299, 1996.

[35]. B. Verma, M. Blumenstein, and S. Kulkarni, "A neural network based technique for data compression," Proceedings of the IASTED International Conference on Modeling and Simulation, Singapore, pp. 12-16, 1997.

[36]. R. Dony, and S. Haykin, "Neural network approaches to image compression," Proceedings of the IEEE, vol. 83, pp. 289-303, 1995.

[37]. S. Carato, "Neural networks for image compression, neural networks advances and applications," editor Glenbe E., Elsevier Science Publishers, pp. 177-198, 1992.

[38]. A. Namphol, "Image compression with a hierarchical neural network," IEEE transactions on Aerospace and Electronic Systems, vol. 32, pp. 327-337, 1996.

[39]. S. Carroto, G. L. Sicuranza and L. Manzo, "Application of ordered codebooks to image coding: neural network for signal processing," IEEE Workshop, pp. 291-300, 1993.

[40]. E. D. Bodt, M. Cottrell and M. Verleysen, "Using the Kohonen algorithm for quick initialization of simple competitive learning algorithm," European Symposium on Artificial Neural Networks, pp. 19-26, 1999.

[41]. S. C. Cheng, W. H. Tsai, "A neural network implementation of the moment-preserving technique and its application to thresholding," IEEE Transactions on Computers, vol. 42, pp. 501-507, 1993.

[42]. A. S. Pandya, R. B. Macy, "Pattern recognition with neural networks in C++," Hoboken, NJ, IEEE Press, 1995.

[43]. E.A. Riskin, L.E. Atlas and S.R. Lay, "Ordered neural maps and their applications to data compression: neural networks for signal processing", IEEE Workshop on Neural Networks, pp. 543-551, 1991.

[44]. O. T. C. Chen, B.J. Sheu, and W. C. Fang, "Image compression using self-organization networks," IEEE Transactions on Circuits and Systems, vol. 4, pp. 480-489, 1994.

[45]. J. Joutsensalo, "Non linear data compression and representation by combining self organizing map and subspace rule," IEEE International Conference on Neural Networks, vol. 2, pp 637-642, 1992.

[46]. T. Kohonen, "Self-organization and associative memory 3rd edition," Springer-Verlag New York Inc., New York, NY, 1989.

[47]. T. Kohonen, "The self-organizing map," Proceedings of the IEEE International Conference on Neural Networks, pp. 1464-1480, 1990.

[48]. G. A. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," vol.21, pp. 77-88, March 1988.

[49]. G. A. Carpenter and S Grossberg, N Markuzon, J. H. Reynolds and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," vol. 3, pp. 698-713, 1992.

[50]. A. C. Izquierdo, J. C. Sueiro and J. A. H. Mendez, "Self-organizing feature maps and their application to digital coding of information," Proceedings of the International Workshop on Artificial Neural Networks, pp. 401-408, 1991.

[51]. C. Amerijckx, M. Verleysen, P. Thissen, J. D. Legat, "Image compression by self organized Kohonen map," IEEE Transactions on Neural Networks, vol. 9, pp. 503-507, 1998.