ULTRASONIC IMAGE COMPRESSION USING SUBBAND HIERARCHICAL BLOCK PARTITIONING

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
ELECTRICAL ENGINEERING
(With Specialization in Measurement and Instrumentation)

By

SRIKANTH VASAMSETTI

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
JUNE, 2010

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation entitled **"ULTRASOUND IMAGE COMPRESSION USING SUBBAND HIERARCHICAL BLOCK PARTITIONING"** submitted in partial fulfilment of the requirement for the award of the Degree of **Master of Technology** in **Electrical Engineering** with specialization in **Measurement and Instrumentation**, in the Department of Electrical Engineering, **Indian Institute of Technology, Roorkee** is an authentic record of my own work carried out from July 2009 to June 2010 under the guidance and supervision of **Dr. Vinod Kumar** (Professor & Head, Electrical Engineering Department , Indian Institute of Technology, Roorkee).

I have not submitted the matter embodied in this report for the award of any other degree or diploma.

Date: June, 2010

Place: Roorkee

(SRIKANTH VASAMSETTI)

CERTIFICATE

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

(Dr. Vinod Kumar)

Professor & Head

Department of Electrical Engineering

Indian Institute of Technology, Roorkee

Roorkee-247667, Uttrakhand, India

ACKNOWLEDGEMENTS

First and the foremost, I would like to thank my guide **Prof. Vinod Kumar** for his valuable advices, guidance, and encouragement and for sharing his broad knowledge. He has been very generous in providing the necessary resources to carry out my research. He is an inspiring teacher, a great advisor, and most importantly a nice person.

I would like to mention my Special thanks to my parents for their endless support and encouragement and for always believing, and helping me to believe, that I can succeed at anything.

My special thanks to tirupathiraju kanumuri, Raghupati Gali for their valuable suggestion and discussions.

Acknowledgement would be incomplete without a word of gratitude to all my friends for their timely help, encouragement and contribution in making this thesis possible.

(SRIKANTH VASAMSETTI)

ABSTRACT

Image compression is fundamental to the efficient and cost-effective use of medical imaging technology and applications such as teleradiology and image archiving. The transmission time reduces drastically due to compression of image. The jpeg, jpeg2000 and SBHP are the well known compression methods.

In the jpeg algorithm, the image has been divided into blocks. The DCT is to be applied on each 8 by 8 block of image to retrieve the coefficients. To represent compressed data, coefficients of each block have been captured in zigzag arder. The reconstructed image from compressed image is to be obtained by applying the reverse process of encoding. By using jpeg algorithm blocking artifacts are there in the reconstructed image. Due to these artifacts reconstructed images seems to be blurred images. To overcome this problem jpeg2000 algorithm has been implemented.

In the jpeg2000 algorithm, discrete wavelet transform is applied to decompose the image into sub-bands at different resolutions. We perform the precincts, scan patter, and compress bit stream operations to represent the compressed data. Reverse process of encoding has been applied on compressed data to obtain the reconstructed image. In this algorithm energy concentration doesn't consider in the same level. In the SBHP, the energy concentrations were available in the both frequency and space.

In the SBHP algorithm, discrete wavelet transform applied to decompose the image into sub-bands at different resolutions. First, the root sub-band has been chosen for checking the significant. The sub-band is significant if any one of the pixel value in the sub-band greater than threshold value. Significant sub-band partitioned into four parts that's called Quadtree partitioning and again checks for significant for each partitioned part. This partitioned process continues on each part until size comes to 1, i.e. one pixel that is to be used for encoding. There are two types of pixels, significant pixel and non significant pixel which is depends on significant condition. A pixel is significant, if the pixel value greater than threshold value otherwise not. The significant pixel encodes as 1 otherwise 0 with sign of pixel 1 for positive and 0 for negative. If root sub-band or any sub-band is not a significant sub band then it included in List of Insignificant Set.

Secondly, all the sub-bands except root sub-band of all levels have been considered as one band and checks for significant. Significant sub-band partitioned into three parts that's called Octave band partitioning and again checks for significant for each partitioned part. This partitioned process continues on each part until size comes to 1, i.e. one pixel that is to be used for encoding. The sub bands which are included in List of Insignificant Set again checks for significant with decreasing previous threshold value by 1.

Similarly, all sub-bands except last level have been considered as one band for further checking of significant.

I applied the three methods (JPEG, JPEG2000, SBHP) on 10 liver ultrasound images and 10 kidney ultrasound images, and I calculated and tabulated MSE, SNR, PSNR and CR for 10 Liver and 10 Kidney ultrasound images. I drew the graph between Mse vs Cr, SNR vs CR, and PSNR vs CR.I also given the output reconstructed images for every input image. From the graphs, I observed that in case of MSE vs CR graphs SBHP method gives better Mse as compared to JPEG & JPEG2000 methods for a given Compression Ratio (CR). In case of SNR vs CR graphs SBHP method gives better SNR as compared to JPEG & JPEG2000 methods for a given Compression Ratio (CR).). In case of PSNR vs CR graphs SBHP method gives better PSNR as compared to JPEG & JPEG2000 methods for a given Compression Ratio (CR).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1.	MSE:	Mean Square Error
2.	PSNR:	Peak Signal to Noise Ratio
3.	SNR:	Signal to Noise Ratio
4.	CR:	Compression Ratio
5.	DCT:	Discrete Cosine Transform
6.	DWT:	Discrete Wavelet Transform
7.	JPEG:	Joint Photographic Experts Group
8.	RMSE:	Root Mean Square Error
9.	1-D:	One Dimensional
10.	2-D:	Two Dimensional
11.	SBHP:	Subband Hierarchical Block Partioning

CHAPTER 1
INTRODUCTION

1.1. Motivation

The easy, rapid, and reliable digital transmission and storage of medical and biomedical images would be a tremendous boon to the practice of medicine. Patients in rural areas could have convenient access to second opinions. Patients readmitted to hospitals could have earlier imaging studies instantly available. Rather than waiting for others to finish with hardcopy films, medical and surgical teams collaborating on patient care could have simultaneous access to imaging studies on monitors throughout the hospital. This long-term digital archiving or rapid transmission is prohibitive without the use of image compression to reduce the file sizes.

For example, a single 2048 × 2048 X-ray image may use 4 megabytes, and transmitting over a telephone line operating at 9600 bits per second (bps) may take one hour, which would be very inefficient. So, if we want to get better performance, we'll have to either increase the band width of the communication channel or apply some compression during transmission.

Furthermore, the situation of narrow band communication can't be totally eliminated in the near future. In many remote country side places, wide-band communication service may be unavailable; in moving vehicles, ships or planes, it is hard to achieve wide-band communication because of the nature of channel (e.g. fading). So, a compression ratio of at least 10:1 is highly required, and better could reach 30:1. Then for the previous example, it means that the image could be transmitted in only a few minutes.

There already have been many very successful works on image compression, and a large variety of algorithms have been proposed. A standard compression algorithm, JPEG, is available which will get good results on most images except when the compression ratio is high. Recently, the wavelet transform was proposed and it can achieve a better compression ratio without increasing computational complexity. [2],[3]

1.2. Literature Review

A typical still image contains a large amount of spatial redundancy in plain areas where adjacent picture elements (pixels) have almost the same values. In addition, still image can contain subjective redundancy, which is determined by properties of human visual system (HVS) [1]. HVS presents some tolerance to distortion depending upon the image content and viewing conditions. The redundancy (both statistical and subjective) can be removed to achieve compression of the image data. The basic measure for the performance of a compression algorithm is compression ratio defined as ratio between original data size and compressed data size. In lossy compression scheme, image compression algorithm should achieve tradeoff between compression ratio and image quality [2]. A standard objective measure of compressed image quality is peak signal to noise ratio (PSNR). For the common case of 8 bits per picture element of input image, PSNR is defined as the ratio of maximum input symbol energy to mean square error (MSE) which evaluates difference between input image and reconstructed image [1].

$$PSNR(dB) = 10\log\left(\frac{255^2}{MSE}\right) \tag{1.1}$$

where $f(x, y)$ is the original image data and $f^*(x, y)$ is the compressed image value and

$$MSE = \frac{1}{N.M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[f(x,y) - f^*(x,y)\right] \tag{1.2}$$

Transform coding is a widely used method of compressing image information. In a transform based compression system two-dimensional images are transformed from the spatial domain to the frequency domain. An effective transform will concentrate useful information into a few of the low frequency transform coefficients. HVS is more sensitive to energy with low spatial frequency than with high spatial frequency. Therefore compression can be achieved by quantizing the coefficients so that important coefficients (low frequency coefficients) are transmitted and remaining coefficients are discarded. Very efficient and popular ways to achieve compression of image data are based on Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT).

Current standards for compression of still (e.g. JPEG (ISO, 1991), JPEG2000 (ISO, 2001)) and moving images (e.g. MPEG-1 (ISO, 1993), MPEG-2 (ISO, 1994)) uses DCT, which represents an image as a superposition of cosine functions with different discrete frequencies. DCT provides excellent energy compaction and a number of fast algorithms exist for calculating the DCT. Most existing compression systems use square DCT blocks of regular size (ISO, 1991), (ISO, 1993), (ISO, 1994). The use of uniformly sized blocks simplified the compression system but does not take into account the irregular shapes within real image. The block-based segmentation of source image is fundamental limitation of the DCT-based compression system [6]. The degradation is known as "blocking effect" and depends on block size.

In recent time, much of the research activities in image coding have been focused on the Discrete Wavelet Transform (DWT) which has become a standard tool in image compression applications because of their data reduction capability [7], [8]. In wavelet compression system the entire image is transformed and compressed as a single data object rather than block by block as in DCT based compression system. It allows a uniform distribution of compression error across the entire image. DWT offers adaptive spatial-frequency resolution (better spatial resolution at high frequencies and better frequency resolution at low frequencies) that is well suited to the properties of HVS. It can provide better image quality than DCT especially on higher compression ratio [9]. But the implementation of the DCT is less expensive than that of the DWT.

The JPEG committee has developed a new, wavelet-based standard for the compression of still images, to be known as JPEG 2000 [12]. JPEG 2000 has many advantages over JPEG, such as better image quality at the same file size, 25-35% smaller file sizes at comparable image quality, good image quality even at very high compression ratios (over 80:1), low complexity option for devices with limited resources, scalable image files, and progressive rendering and transmission through a layered image file structure.

The main disadvantage with JPEG 2000 is that it does not consider the energy concentration in the same level. This is done by Subband Hierarchical Block Partitioning (SBHP) algorithm. It exploits two fundamental characteristics of an image transform—the well defined hierarchical structure, and energy clustering in frequency and in space and encodes with very low complexity. So the objective of my dissertation is to implement JPEG, JPEG2000 and SBHP methods, and compare the results of SBHP with JPEG and JPEG 2000. SBHP is a block based image coding method which considers the clustering of energy in frequency and space in hierarchical structures of transformed image. [23]

1.3. Organization Of Thesis

Chapter 1 discusses motivation for my thesis work and different methods available in the literature.

Chapter 2 discusses the basic concepts of Image Compression.

Chapter 3 discusses the basic concepts of JPEG Algorithm.

Chapter 4 discusses the basic concepts of JPEG 2000 Algorithm.

Chapter 5 discusses the basic concepts of SBHP Algorithm.

Chapter 6 presents the results and comparative study between different methods.

Chapter 7 gives conclusions and future scope.

Chapter 2

IMAGE COMPRESSION

2.1. INTRODUCTION

In the field of image processing, image compression is the current topic of research. Image compression plays a crucial role in many important and diverse applications, including televideo conferencing, remote sensing, document & medical and facsimile transmission.

2.2. Need For Compression

Image data is by its nature multidimensional and tend to take up a lot of space

- Pictures take up a lot of storage space (either disk or memory).
- A 1000x1000 picture with 24 bits per pixel takes up 3 megabytes.
- The Encyclopedia Britannica scanned at 300 pixels per inch and 1 bit per pixel requires 25,000 pages $\times$1,000,000 bytes per page = 25 gigabytes.
- Applications: HDTV, film, remote sensing and satellite image transmission, network communication, image storage, medical image processing, fax.

2.3. Principles Behind Compression

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image/video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS). In general, three types of redundancy can be identified:

- **Spatial Redundancy** or correlation between neighboring pixel values.
- **Spectral Redundancy** or correlation between different color planes or spectral bands.
- **Temporal Redundancy** or correlation between adjacent frames in a sequence of images (in video applications).

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. Since we will focus only on still image compression, we will not worry about temporal redundancy. Different methods for redundancy reduction are

- **Spatial redundancy**: DCT, DWT, DPCM
- **Statistical redundancy**: Run-Length coding, Variable-Length coding

2.4. Image Compression Model

A typical image compression model consists of source encoder which is resposible for reducing or eliminating any coding, interpixel or psycho visual redundancies in the input image. Channel is a transmission path and source decoder reconstructs the original image whose function is opposite to that of source encoder. The figure.1 shows the block diagram of image compression model [1].

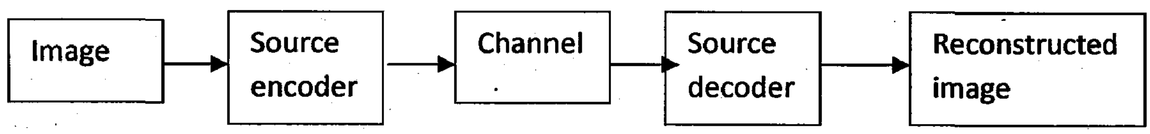


Fig 2.1 Image compression model [1]

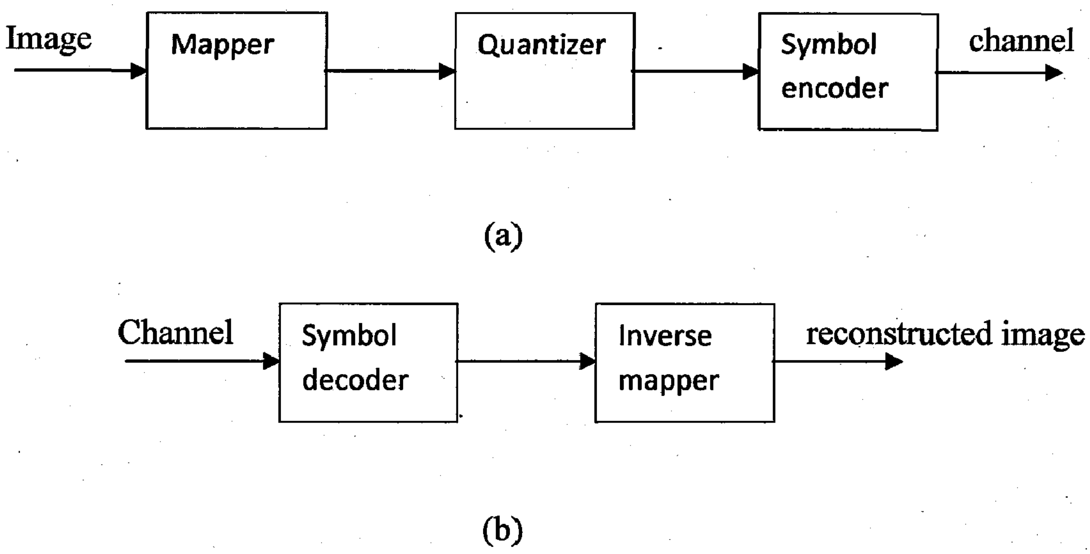


Fig 2.2 (a) Source encoder (b) Source decoder [1]

The source encoder consists of three blocks. The first stage of the source encoding process, the mapper transforms the input data into a format designed to reduce inter pixel redundancies in the input image. This operation is generally reversible and may or may not reduce directly the amount of data required to represent the image.

- The second stage, or quantizer block in figure.2(a), reduces the accuracy of the mappers output in accordance with some pre established fidelity criterion. The stage reduces the psycho visual redundancies of the input image. This operation is irreversible. Thus it must be omitted when error free compression is desired.
- In the third and final stage of the source encoding process, the symbol coder block in figure.2 (a) creates a fixed- or variable-length code to represent the quantizer output and maps the output in accordance with the code.
- The source decoder shown in figure.2 (b) contains only two components symbol decoder and an inverse mapper. These blocks perform, in reverse order, the inverse operations of the source encoder's, symbol encoder and mapper blocks.

2.5. LOSS LESS COMPRESSION METHODS

In numerous applications error-free compression is the only acceptable means of data compression. One such application is the archival of medical or business documents, where lossy compression usually is prohibited for legal reasons. Another is the processing of satellite image, where both the use and cost of collecting the data makes any loss undesirable. Yet another is digital radiography, where the loss of information can compromise diagnostic accuracy. In these and other cases, the need for error –free compression is motivated by the intended use or nature of the image under consideration. The lossless method normally provides compression ratios of 2 to 10. [4]

2.5.1 Run Length Coding [4], [5]

This method reduces only inter pixel redundancy. The following example illustrates the Run length coding method.

Original Image

63	63	63	63	64	64	64	78	89	89	89	89

Compressed Image

63, 4,64,3,78,1,89,4

- Code the number of pixels taking the same value along a given scan line.
- Works particularly well on binary images since only length of run needs to encoded.

- Works by utilizing scan line coherence.
- Bit-plane run length encoding is used on non-binary images by considering each bit of the, say 8 bit, image one at a time.
- Compression rates of 1.5:1 (gray-scale / color images), 4:1 (binary images) and 2:1 (bit-plane compression on gray-scale /color images)
- May cause a data explosion: the final file may be larger than the original one.

2.5.2 Huffman Coding [4], [5]

- This is the most popular technique for removing coding redundancy.
- Huffman coding works on the image brightness histogram.
- Finds the most commonly occurring brightness patterns and uses the shortest codes to represent these.
- Compression rates of 1.5 to 2:1.

Huffman coding may also be used after run length coding to give further compression.

2.5.3 Arithmetic Coding [4], [5]

Arithmetic coding is a lossless coding method which does not suffer from the mentioned drawbacks and which tends to achieve a higher compression ratio than Huffman coding. However, Huffman coding can generally be realized with simpler software and hardware.

The basic idea behind arithmetic coding is to map the input sequence of symbols into one single codeword. Symbol blocking is not needed since the codeword can be determined and updated incrementally as each new symbol is input (symbol-by-symbol coding). At any time, the determined codeword uniquely represents all the past occurring symbols. Although the final codeword is represented using an integral number of bits, the resulting average number of bits per symbol is obtained by dividing the length of the codeword by the number of encoded symbols.

Arithmetic Coding Algorithm:

1) Divide the interval [0,1] into segments corresponding to the M symbols; the segment of each symbol has a length proportional to its probability.

2) Choose the segment of the first symbol in the string message.

3) Divide the segment of this symbol again into M new segments with length proportional to the symbols probabilities.

4) From these new segments, choose the one corresponding to the next symbol in the message.

5) Continue steps 3) and 4) until the whole message is coded.

6) Represent the segment's value by a binary fraction.

2.5.4 Predictive Coding [1],[5]

Original Image

63	63	63	63	64	64	64	78	89	89	89	89

Compressed Image

63,0,0,0,1,0,0,14,11,0,0,0

- Stores the deference between successive pixels' brightness in fewer bits.
- Relies on the image having smooth changes in brightness: at sharp changes in the image we need overload patterns.

2.6 Performance Evaluation Parameters [8], [9]

To compare different algorithms of lossy compression several approaches of measuring the loss of quality have been devised. In the MI context, where the ultimate use of an image is its visual assessment and interpretation, subjective and diagnostic evaluation approaches are the most appropriate. However, these are largely dependent on the specific task at hand and moreover they entail costly and time-consuming procedures. In spite of the fact that they are often inadequate in predicting the visual (perceptual) quality of the decompressed image, objective measures are often used since they are easy to compute and are applicable to all kinds of images regardless of the application.

Compression ratio is defined as the nominal bit depth of the original image in bits per pixel (bpp) divided by the bpp necessary to store the compressed image. For each compressed and reconstructed image, an error image was calculated. From the error data, maximum absolute error (MAE), mean square error (MSE), root mean square error (RMSE), signal to noise ratio (SNR), and peak signal to noise ratio (PSNR) were calculated.

The maximum absolute error (MAE) is calculated as

$$MAE = \max\left|f(x,y) - f^*(x,y)\right| \qquad (2.1)$$

where $f(x, y)$ is the original image data and $f^*(x, y)$ is the compressed image value. The formulae for calculated image matrices are:

$$MSE=\frac{1}{N.M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[f(x,y)-f^{*}(x,y)\right]^{2} \tag{2.2}$$

$$RMSE=\sqrt{MSE} \tag{2.3}$$

Where M and N are the matrix dimensions in x and y, respectively.

$$SNR=10\log\left\{\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}f(x,y)^{2}}{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[f(x,y)-f^{*}(x,y)\right]}\right\} \tag{2.4}$$

$$PSNR=20\log\left(\frac{255}{RMSE}\right) \tag{2.5}$$

2.7 Interim Conclusions

Lossless methods gives very low compression ratios but, they reproduce the original image without loss of any information. A lossless method is particularly used for tele-medicine applications. Arithmetic coding, gives the best compression rate, but are associated with larger compression and decompression times for images [6]. With more local variation in pixel values, they may enlarge images instead of compressing them. The Adaptative Huffman method gives the best compromise between compression rate and computing times [6]. Arithmetic coding achieves higher compression than Huffman coding. However, Huffman coding can generally be realized with simpler software and hardware [2].

Lossy compression methods are used when high compression ratios are required at the cost of loss of some useful information. The JPEG (dct based) can be used up to compression ratios less than 52 [7]. This is a standard method used in practice because it takes very less time for both encoding and decoding. The main limitation of this method is the block artefacts increases as we go for more compression ratios [8]. Wavelet based methods takes medium time for both encoding and decoding and there will be no block artefacts. The JPEG 2000 (based on wavelets) is the current standard for image compassion.

Chapter 3

THE JPEG CODING

3.1. Introduction

The important feature of the DCT is that it takes correlated input data and concentrates its energy in just the first few transform coefficients. If the input data consists of correlated quantities, then most of the n transform coefficients produced by the DCT are zeros or small numbers, and only a few are large (normally the first ones). The early coefficients contain the important (low-frequency) image information and the later coefficients contain the less-important (high-frequency) image information. Compressing data with the DCT is therefore done by quantizing the coefficients. The small ones are quantized coarsely (possibly all the way to zero), and the large ones can be quantized finely to the nearest integer. After quantization, the coefficients (or variable-size codes assigned to the coefficients) are written on the compressed stream. Decompression is done by performing the inverse DCT on the quantized coefficients. This results in data items that are not identical to the original ones but are not much different. The DCT is applied to small parts (data blocks) of the image. It is computed by applying the DCT in one dimension to each row of a data block, then to each column of the result [14], [15], [16], [17]. Because of the special way the DCT in two dimensions is computed, we say that it is separable in the two dimensions. Because it is applied to blocks of an image, we term it a "blocked transform." It is defined by

$$G_{ij} = \sqrt{\frac{2}{m}}\sqrt{\frac{2}{n}}C_i C_j \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} p_{xy} \cos\left[\frac{(2y+1)j\pi}{2m}\right]\cos\left[\frac{(2x+1)i\pi}{2n}\right] \tag{3.1}$$

For $0 \leq i \leq n\text{-}1$ and $0 \leq j \leq m-1$ and for C_i and C_j defined by Equation G_f .The first coefficient G_{00} is termed the "DC coefficient," and the remaining coefficients are called the "AC coefficients." The image is broken up into blocks of $n \times m$ pixels p_{xy} (with $n = m = 8$ typically), and Equation G_{ij} is used to produce a block of $n \times m$ DCT coefficients G_{ij} for each block of pixels. The coefficients are then quantized, which results in lossy but highly efficient compression. The decoder reconstructs a block of quantized data values by computing the IDCT whose definition is

$$p_{xy} = \sqrt{\frac{2}{m}}\sqrt{\frac{2}{n}}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} C_i C_j G_{ij} \cos\left[\frac{(2x+1)i\pi}{2n}\right]\cos\left[\frac{(2y+1)j\pi}{2m}\right] \tag{3.2}$$

where $C_f = \left\{ \begin{matrix} \frac{1}{\sqrt{2}}, & f=0 \\ 1, & f>0, \end{matrix} \right\}$, for $0 \le x \le n - 1$ and $0 \le y \le m - 1$.

Steps involved in DCT image compression technique:

1. The image is divided into k blocks of 8×8 pixels each. The pixels are denoted by p_{xy}. If the number of image rows (columns) is not divisible by 8, the bottom row (rightmost column) is duplicated as many times as needed.

2. The DCT in two dimensions is applied to each block B_i. The result is a block (we'll call it a vector) $W^{(i)}$ of 64 transform coefficients $w_j^{(i)}$ (where $j = 0, 1, \ldots, 63$). The k vectors $W^{(i)}$ become the rows of matrix **W**

$$\mathbf{W} = \begin{bmatrix} w_0^{(1)} & w_1^{(1)} & \ldots & w_{63}^{(1)} \\ w_0^{(2)} & w_1^{(2)} & \ldots & w_{63}^{(2)} \\ \vdots & \vdots & & \\ w_0^{(k)} & w_1^{(k)} & \ldots & w_{63}^{(k)} \end{bmatrix} \tag{3.3}$$

3. The 64 columns of **W** are denoted by $C^{(0)}, C^{(1)}, \ldots, C^{(63)}$. The k elements of $C^{(j)}$ are $\left(w_j^{(1)}, w_j^{(2)}, \ldots, w_j^{(k)}\right)$. The first coefficient vector $C^{(0)}$ consists of the k DC coefficients.

4. Each vector $C^{(j)}$ is quantized separately to produce a vector $Q^{(j)}$ of quantized coefficients. The elements of $Q^{(j)}$ are then written on the compressed stream. In practice, variable-size codes are assigned to the elements, and the codes, rather than the elements themselves, are written on the compressed stream.

3.2 Overview of JPEG Image Compression

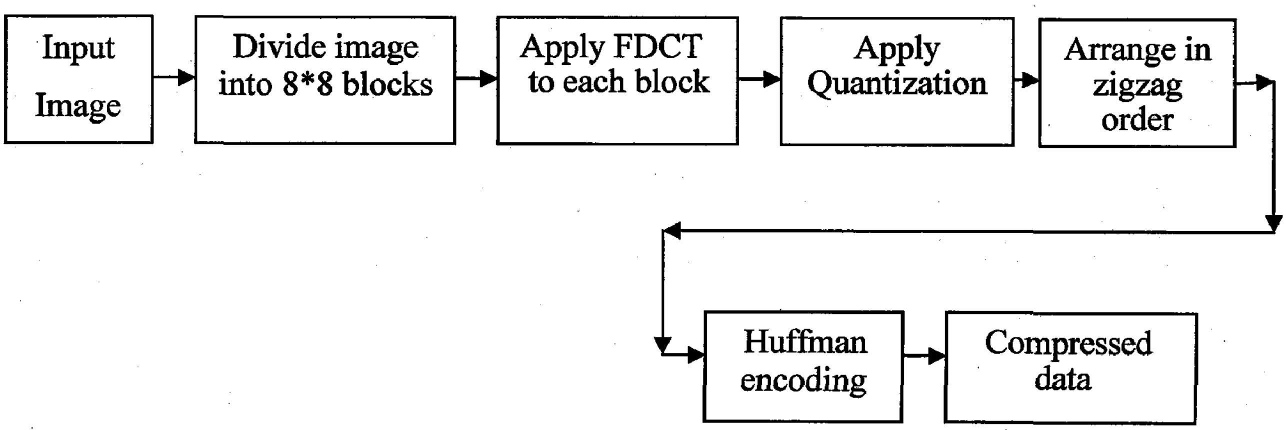


Fig 3.1 Block diagram of JPEG Encoder [15]

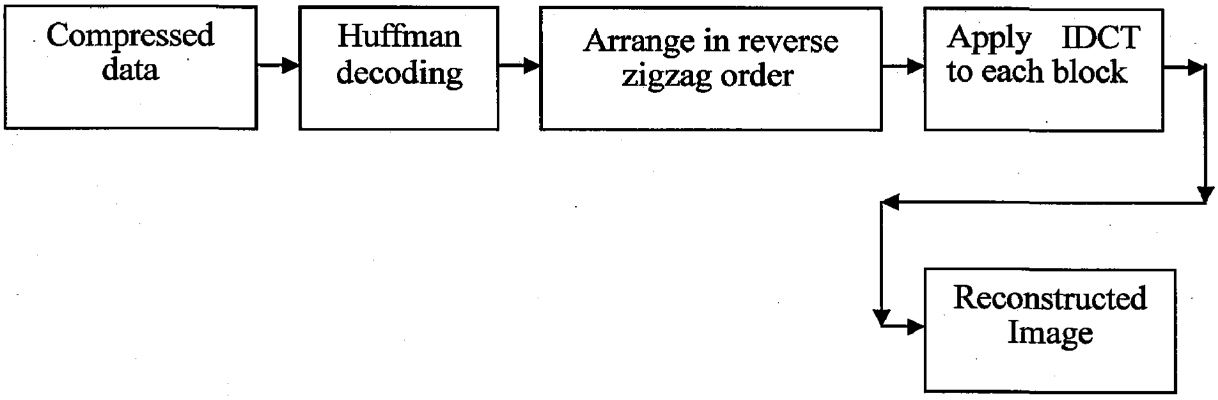


Fig 3.2 Block diagram of JPEG Decoder [15]

3.3 JPEG Encoding:

1. Divide into 8*8 matrices:

The image is broken into 8*8 blocks of pixels.

2 Apply FDCT:

Each block is transformed by the forward DCT (FDCT) into a set of 64 values referred to as DCT coefficients. One of these values is referred to as the DC coefficient and the other 63 as the AC coefficients.

3. Apply Quantization:

After output from the FDCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a 64-element Quantization Table. The purpose of quantization is to achieve further compression by representing DCT coefficients with no greater precision than is necessary to achieve the desired image quality. Quantization is defined as division of each DCT coefficient by its corresponding quantizer step size, followed by rounding to the nearest integer:

$$F^{Q}(u, v) = \text{Integer Round}\left(\frac{F(u,v)}{Q(u,v)}\right) \quad (3.4)$$

4. Zigzag sequence:

All of the quantized coefficients are ordered into the "zigzag" sequence, as shown in Figure below. This ordering helps to facilitate entropy coding by placing low-frequency coefficients (which are more likely to be nonzero) before high-frequency coefficients.

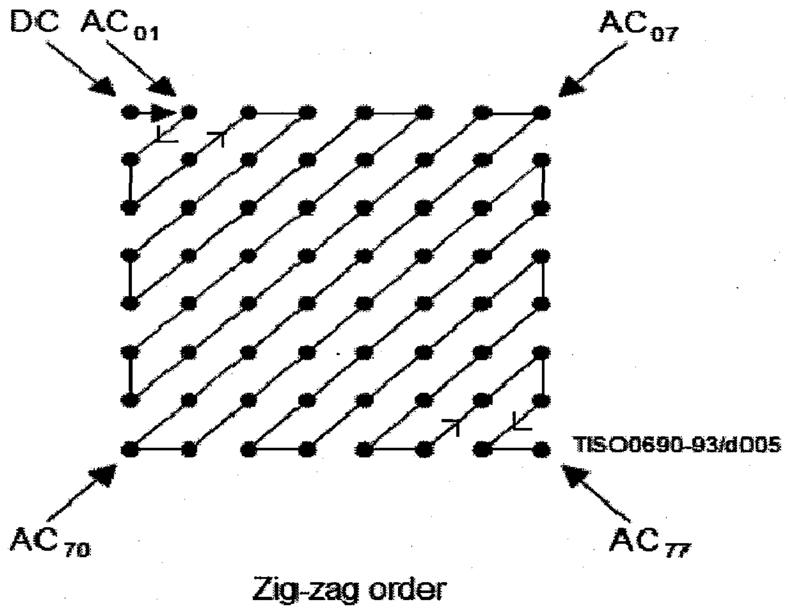


Fig 3.3 Zigzag order [16]

5. Huffman encoding:

Arrange matrix into row vector & encode the final row vector which is one dimensional using Huffman encoding to further reduce redundancy and hence increasing compression ratio.

3.4 JPEG Decoding:

1. Huffman decoding:

Apply Huffman decoding to the received data and arrange in matrix.

2. Reverse zigzag ordering:

Divide matrix into 8*8 blocks and arrange reverse zigzag order.

3. Apply IDCT: Apply inverse discrete transform on the image reconstruct the matrix.

3.5 Interim conclusions

The JPEG algorithm yields good results for compression ratios of 10:1 and below (on 8-bit gray-scale images), but at higher compression ratios the underlying block nature of the transform begins to show through the compressed image. By the time compression ratios have reached 24:1, only the DC (lowest frequency) coefficient is getting any bits allocated to it, and the input image has been approximated by a set of 8×8 blocks. Consequently, the decompressed image has substantial blocking artifacts for medium and high compression ratios. This disadvantage is eliminated by using JPEG 2000 method.

chapter4

JPEG2000 CODING

4.1. Introduction

The data compression field is very active, with new approaches, ideas, and techniques being developed and implemented all the time. JPEG is widely used for image compression but is not perfect. The use of the DCT on 8×8 blocks of pixels results sometimes in a reconstructed image that has a blocky appearance (especially when the JPEG parameters are set for much loss of information). This is why the JPEG committee has developed a new, wavelet-based standard for the compression of still images, to be known as JPEG 2000. JPEG 2000 has many advantages over JPEG, such as better image quality at the same file size, 25-35% smaller file sizes at comparable image quality, good image quality even at very high compression ratios (over 80:1), low complexity option for devices with limited resources, scalable image files, and progressive rendering and transmission through a layered image file structure. [19]

JPEG 2000 is not only intended to provide rate-distortion and subjective image quality performance superior to existing standards, but also to provide features and functionalities that current standards can either not address efficiently or in many cases cannot address at all. Lossless and lossy compression, embedded lossy to lossless coding, progressive transmission by pixel accuracy and by resolution, robustness to the presence of bit-errors and region-of-interest coding, are some representative features. It is interesting to note that JPEG2000 is designed to address the requirements of a diversity of applications, e.g. Internet, color facsimile, printing, scanning, digital photography, remote sensing, mobile applications, medical imagery, digital library and E-commerce.

JPEG-2000 has a long list of features, a subset of which are [11], [19].[20]

- High compression efficiency. Bitrates of less than 0.25 bpp are expected for highly detailed greyscale images.
- The ability to handle large images, up to 232×232 pixels (the original JPEG can handle images of up to 216×216).
- Progressive image transmission. The proposed standard can decompress an image progressively by SNR, resolution, colour component, or region of interest.

- Easy, fast access to various points in the compressed stream.
- The decoder can pan/zoom the image while decompressing only parts of it.
- The decoder can rotate and crop the image while decompressing it.
- Error resilience. Error-correcting codes can be included in the compressed stream, to improve transmission reliability in noisy environments.

One of the new, important approaches to compression introduced by JPEG 2000 is the "compress once, decompress many ways" paradigm. The JPEG 2000 encoder selects a maximum image quality Q and maximum resolution R, and it compresses an image using these parameters. The decoder can decompress the image at any image quality up to and including Q and at any resolution less than or equal to R. Suppose that an image I was compressed into B bits. The decoder can extract A bits from the compressed stream (where $A < B$) and produce a lossy decompressed image that will be identical to the image obtained if I was originally compressed lossy to A bits.

In general, the decoder can decompress the entire image in lower quality and/or lower resolution. It can also decompress parts of the image (regions of interest) at either maximum or lower quality or resolution. Even more, the decoder can *extract* parts of the compressed stream and assemble them to create a new compressed stream without having to do any decompression. Thus, a lower-resolution and/or lower-quality image can be created without the decoder having to decompress anything. The advantages of this approach are (1) it saves time and space and (2) it prevents the build up of image noise, common in cases where an image is lossy compressed and decompressed several times.

Features of JPEG2000:

1. Superior low bit-rate performance: below 0.25 bpp for highly detailed gray scale images.
2. Continuous-tone compression: various dynamic ranges (e.g., 1 to 16 bits).
3. The ability to handle large images, up to 232×232 pixels (the original JPEG can handle images of up to 216×216).
4. Lossless and lossy compression: with embedded bit stream and allow progressive lossy to lossless build up.
5. Easy, fast access to various points in the compressed stream.
6. Region-of-interest (ROI) coding.

7. Robustness to bit errors.

8. Error resilience: Error-correcting codes can be included in the compressed stream, to improve transmission reliability in noisy environments.

9. Protective image security (watermarking, labelling, stamping, or encryption).

4.1.1 Discrete Wavelet Transform

Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. The basic idea of the wavelet transform is to represent any arbitrary signal 'X' as a superposition of a set of such wavelets or basis functions. These basis functions are obtained from a single photo type wavelet called the mother wavelet by dilation (scaling) and translation (shifts).discrete wavelet transform for 1-dimension is given below.

$$w(a,b) = \frac{1}{\sqrt{a}}\Psi\left(\frac{X-b1}{a}\right) \qquad (4.1)$$

The indexes $w(a,b)$ are called wavelet coefficients of signal X, and a is dilation and b is translation, Ψ is the transforming function, which is known as mother wavelet. Low frequencies are examined with low temporal resolution while high frequencies with more temporal resolution. A wavelet transform combines both low pass and high pass filtering in spectral decomposition of signals.

4.1.2 2 D Wavelet Analysis

Images are treated as two dimensional signals, they change horizontally and vertically, thus 2D wavelet analysis must be used for images. 2D wavelet analysis uses the same 'mother wavelets' but requires an extra step at every level of decomposition. The 1D analysis filtered out the high frequency information from the low frequency information at every level of decomposition; so only two subsignals were produced at each level. In 2D, the images are considered to be matrices with N rows and M columns. At every level of decomposition the horizontal data is filtered, and then the approximation and details produced from this are filtered on columns.

At every level, four sub-images are obtained; the approximation, the vertical detail, the horizontal detail and the diagonal detail. Below the Saturn image has been decomposed to one level. The wavelet analysis has found how the image changes vertically, horizontally and diagonally.

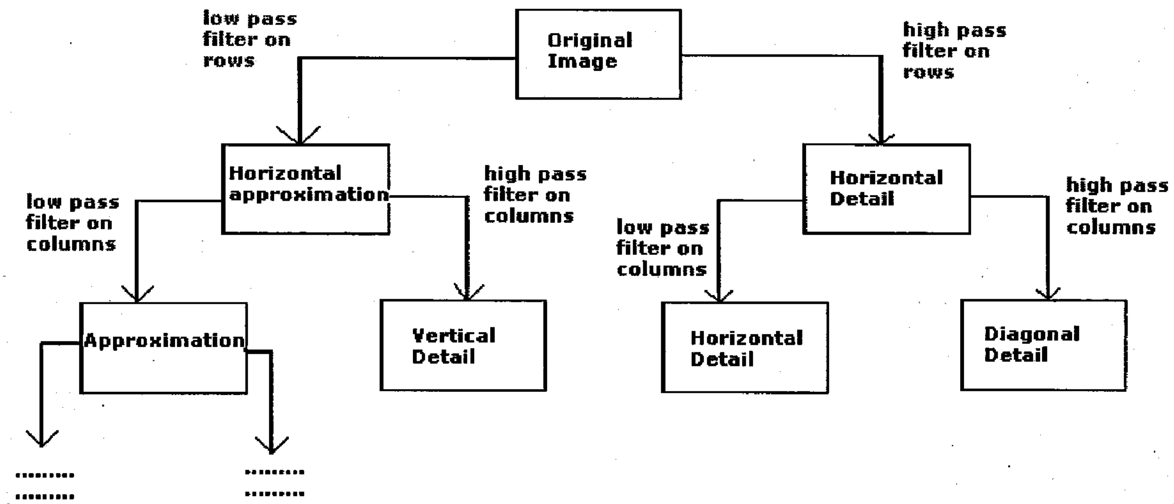


Fig 4.1 Sub band decomposition of 2-D image [20]

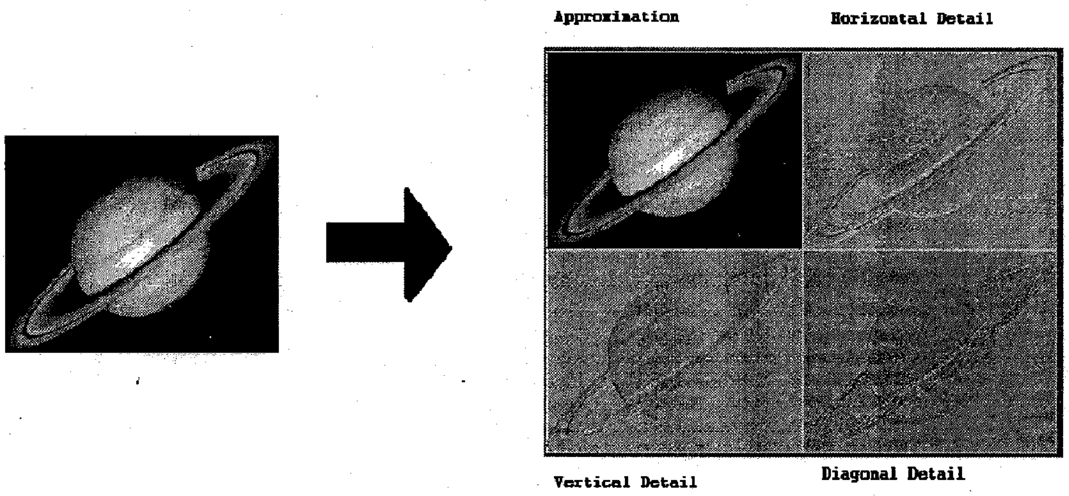


Fig.4.2. 2-D Decomposition of Saturn Image to level 1 [20]

Actually no compression is achieved by wavelets. It decomposes the image different frequency bands and actual compression is done by quantisation and entropy coding. The quantization and encoding steps are the crucial aspects of wavelet transform compression because they are the cause for actual compression.

It has been known throughout the research community for several years that the wavelet transform is superior to DCT methods in image compression. Thus, in March of 2000, JPEG published the JPEG2000 standard based on wavelet technology. The compression method of JPEG2000 is similar to that of JPEG. However, JPEG2000 uses the wavelet transform instead of the block-based DCT. This allows for the user to specify the size of the processing block (small block sizes reduce the memory requirement while large block sizes improve compression gain and reconstructed image quality). After transformation, coefficients are quantized and encoded as in the JPEG standard. The JPEG2000 standard promises a 20%-25% smaller average file size with comparable quality than the original JPEG standard [22].

The JPEG2000 standard provides a set of features that are of vital importance to many high-end and emerging applications, by taking advantage of new technologies. It addresses areas where current standards fail to produce the best quality or performance and provides capabilities to markets that currently do not use compression. The markets and applications better served by the JPEG2000 standard are Internet, color facsimile, printing, scanning (consumer and pre-press), digital photography, remote sensing, mobile, medical imagery, digital libraries / archives and E-commerce. Each application area imposes some requirements that the standard should fulfil. The main features that this standard possesses are: superior low bit-rate performance, continuous-tone and bi-level compression, lossless and lossy compression, progressive transmission by pixel accuracy and resolution, random code stream access and processing, robustness to bit-errors.

4.2. JPEG2000 Coding Overview

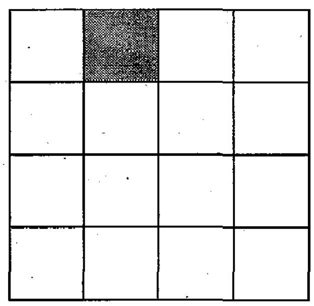
The main components of JPEG2000 are shown in fig 4.3 below.

1. Tiling:

The image is divided into rectangular blocks of fixed size called as tiles.

2. Image Transformation:

2-dimensional Wavelet transforms are applied to each tile of the image and decomposed into multilevel.



Tiling

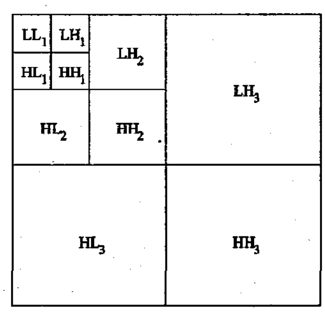
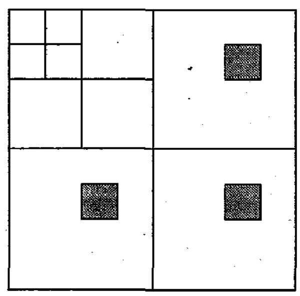
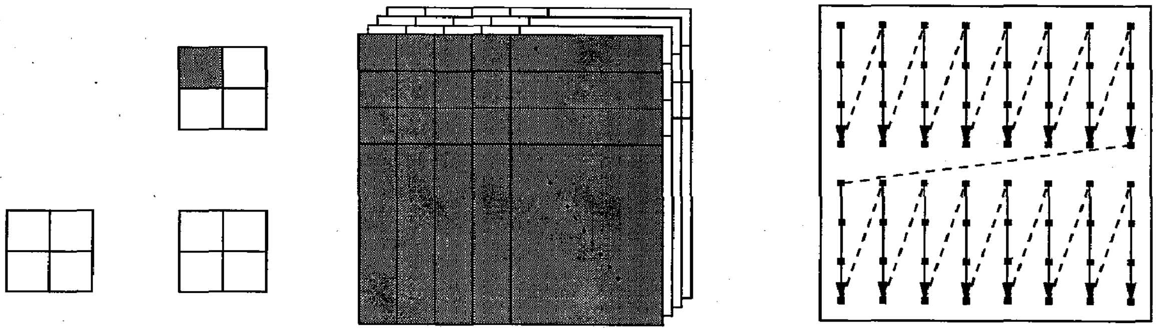


Image transform



precinct



Code block Bit plane Scan pattern

Fig 4.3 Main components of JPEG 2000 [24]

3. Precinct:

The image transform in each tile is divided into rectangular blocks called precinct.

4. Code block:

Each precinct is further divided into non overlapping rectangular blocks of equal size called code blocks.

5. Bit plane:

Each pixel in the code block are converted to binary form. The MSB of all the pixels in the code block represents most significant bit plane and LSB represents least significant bit plane.

6. Scan pattern:

The bits in each bit plane are arranged in raster scan pattern where the bits are scanned row wise 4 bits and then moves to next column and so on as shown in fig 4.3.

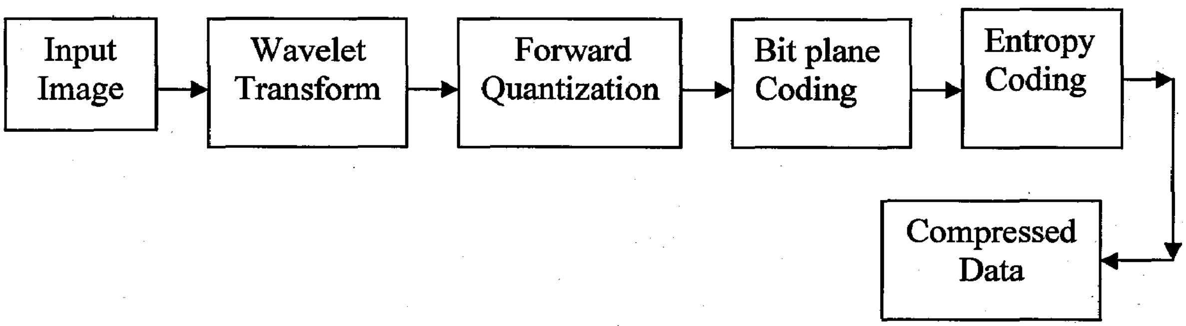


Fig 4.4 Block Diagram JPEG 2000 Encoder [19]

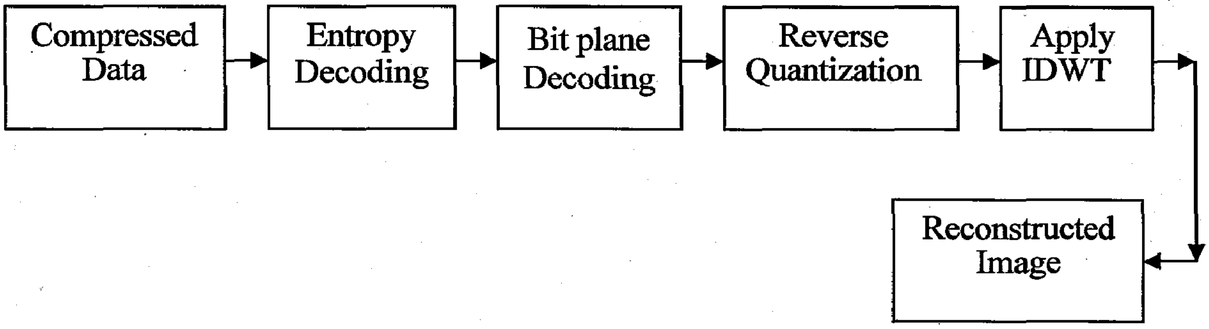


Fig 4.5 Block Diagram JPEG 2000 Decoder [19]

4.3 JPEG2000 Encoding:

Steps:

1. Tiles:

The image components can be partitioned into tiles, which are rectangular non-overlapping blocks, and thus creating tile components that can be compressed independently of each other.

2. Wavelet Transform:

Wavelet transform may be performed on the tile components. The wavelet transform decomposes the tile-components into different decomposition levels, each of which contains a number of sub bands filled with transform coefficients.

3. Quantization:

Before entering into the entropy coding phase, the quantization process is carried out to reduce the precision of the transform coefficients. Note that for the lossless case, the quantizer is set to one, i.e. no loss in precision.

4. Precincts and Code blocks:

Each Subband, after quantization, is divided into non overlapping rectangular blocks, called code blocks, of equal size. The dimensions of the code blocks are powers of 2 (e.g., of size 16 x 16 or 32 x 32) and the total number of coefficients in a code block should not exceed 4096. The code blocks formed by the quantizer indexes corresponding to the quantized wavelet coefficients constitute the input to the entropy coder. Collections of spatially consistent code locks taken from each Subband at each resolution level are called precincts and will form a packet partition in the bit stream structure. The purpose of precincts is to enable spatially progressive bit streams. [11]

5. Bit plane Coding:

The coefficients in a code block are separated into bit-planes. The individual bit-planes are coded in 1-3 coding passes. Consider a quantized code-block to be an array of integers in sign-magnitude representation, and then consider a sequence of binary arrays with one bit from each coefficient. The first such array contains the most significant bit (MSB) of all the magnitudes. The second array contains the next MSB of all the magnitudes, continuing in this fashion until the final array which consists of the least significant bits of all the magnitudes. These binary arrays are referred to as bit planes. Starting from the first bit plane having at least a single 1, each bit plane is encoded in three passes (referred to as sub-bit planes). The scan pattern followed for the coding of bit planes, within each code-block (in all sub bands). This scan pattern is basically a column-wise raster within stripes of height four. At the end of each stripe, scanning continues at the beginning (top-left) of the next stripe, until an entire bit plane (of a code block) has been scanned.

The prescribed scan is followed in each of the three coding passes. The decision as to which pass a given bit is coded in is made based on the "significance" of that bit's location and the significance of neighbouring locations. A location is considered significant if a 1 has been coded for that location (quantized coefficient) in the current or previous bit planes. The first pass in a new bit plane is called the significance propagation pass. A bit is coded in this pass if its location is not significant, but at least one of its eight-connected neighbours is significant. If a bit is coded in this pass, and the value of that bit is 1, its location is marked as significant for the purpose of coding subsequent bits in the current and subsequent bit planes. Also, the sign bit is coded immediately after the 1 bit just coded. The second pass is the magnitude refinement pass. In this pass, all bits from locations that became significant in a previous bit plane are coded. The third and final pass is the clean-up pass, which takes care of any bits not coded in the first two passes.

6. Entropy Coding:

Huffman encoding is used for this entropy coding. Huffman encoding to further reduce the redundancy and hence increasing compression ratio. [20]

4.4 JPEG2000 Decoding:

The decoding process is basically the reverse of the encoding one.

Steps:

1. Huffman decoding:

   Apply Huffman decoding to the received data and arrange in matrix.

2. Bit Plane Decoding:

   This is reverse process of bit plane encoding.

3. Reverse Quantization:

   Apply the inverse Quantization on the obtained data.

4. Inverse wavelet Transform:

   Apply the inverse wavelet transform on the reverse Quantization data.

4.5 Interim conclusions

JPEG is widely used for image compression but is not perfect. The use of the DCT on 8×8 blocks of pixels results sometimes in a reconstructed image that has a blocky appearance (especially when the JPEG parameters are set for much loss of information). This is why the JPEG committee has developed a new, wavelet-based standard for the compression of still images, to be known as JPEG 2000. JPEG 2000 has many advantages over JPEG, such as better image quality at the same file size, 25-35% smaller file sizes at comparable image quality, good image quality even at very high compression ratios, low complexity option for devices with limited resources, scalable image files, and progressive rendering and transmission through a layered image file structure. JPEG2000 is that it does not consider the energy concentration in the same level. This is done by SUBBAND HIERARCHICAL BLOCK PARTIONING (SBHP) algorithm.

Chapter 5
THE SBHP CODING

5.1. Introduction

The main disadvantage with JPEG2000 is that it does not consider the energy concentration in the same level. This is done by SUBBAND HIERARCHICAL BLOCK PARTIONING (SBHP) algorithm. It exploits two fundamental characteristics of an image transform—the well defined hierarchical structure, and energy clustering in frequency and in space and encodes with very low complexity.

Consider an image ***x*** which has been adequately transformed using an appropriate Subband transformation (most commonly, the discrete wavelet transform). The transformed image is said to exhibit a hierarchical pyramidal structure defined by the levels of decomposition, with the topmost level being the root. The finest pixels lie at the bottom level of the pyramid while the coarsest pixels lie at the top (root) level. Following the ideas of SPIHT, we say that a set ***T*** of pixels is *significant* with respect to ***n*** if

$$\max_{(i,j)\in T}\left\{\left|c_{i,j}\right|\right\} \geq 2^n \qquad (5.1)$$

Otherwise it is *insignificant*. We can write the significance of a set ***T*** as

$$\tau_n(T) = \begin{cases} 1, & \text{if } 2^n \leq \max_{(i,j)\in T}\left\{\left|c_{i,j}\right|\right\} < 2^{n+1} \\ 0, & \text{else} \end{cases} \qquad (5.2)$$

The SBHP algorithm makes use of rectangular regions of image. These regions or sets, henceforth referred to as sets of type ***S***, can be of varying dimensions. The dimension of a set ***S*** depends on the dimension of the original image and the Subband level of the pyramidal structure at which the set lies. The other types of sets used in the SBHP algorithm are referred to as sets of type ***I***. These sets are obtained by chopping off a small square region from the top left portion of a larger square region. A typical set ***I*** is illustrated. A set ***I*** is always decomposed into ***S*** sets in a prescribed way, so as to progress through the transformed image from coarser to finer resolution sub bands. The coding part of SBHP always takes place on the ***S*** sets.

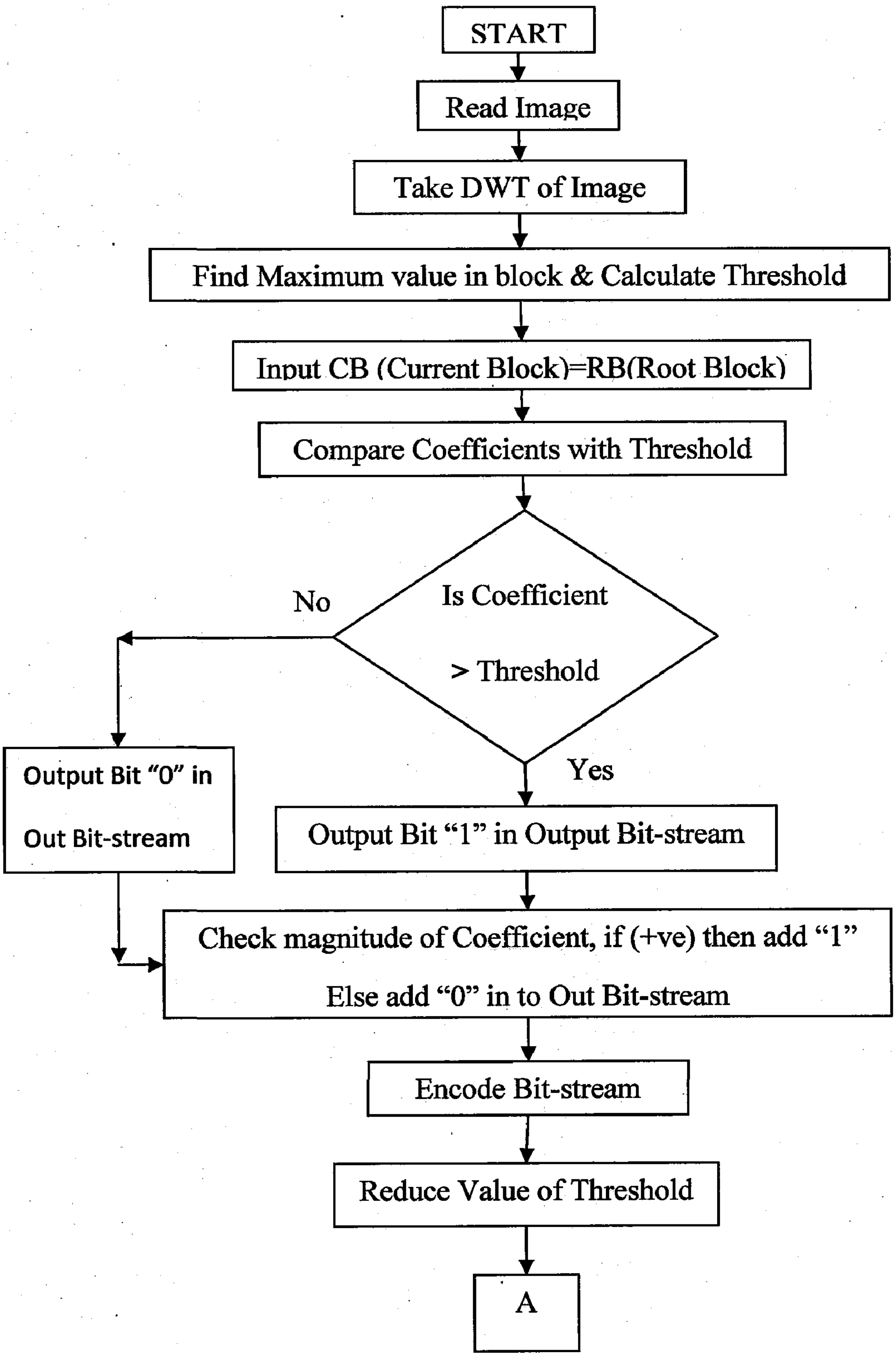
To encode an ***S*** set, the SBHP algorithm follows closely the methodology used in the SPIHT algorithm. The difference lies in the sorting pass where instead of using spatial orientation trees for significance testing, we use sets of type ***S*** as defined above. The motivation behind this is to exploit the clustering of energy found in transformed images and concentrate on those areas of the set which have high energy. This ensures that pixels with high information content are coded first.

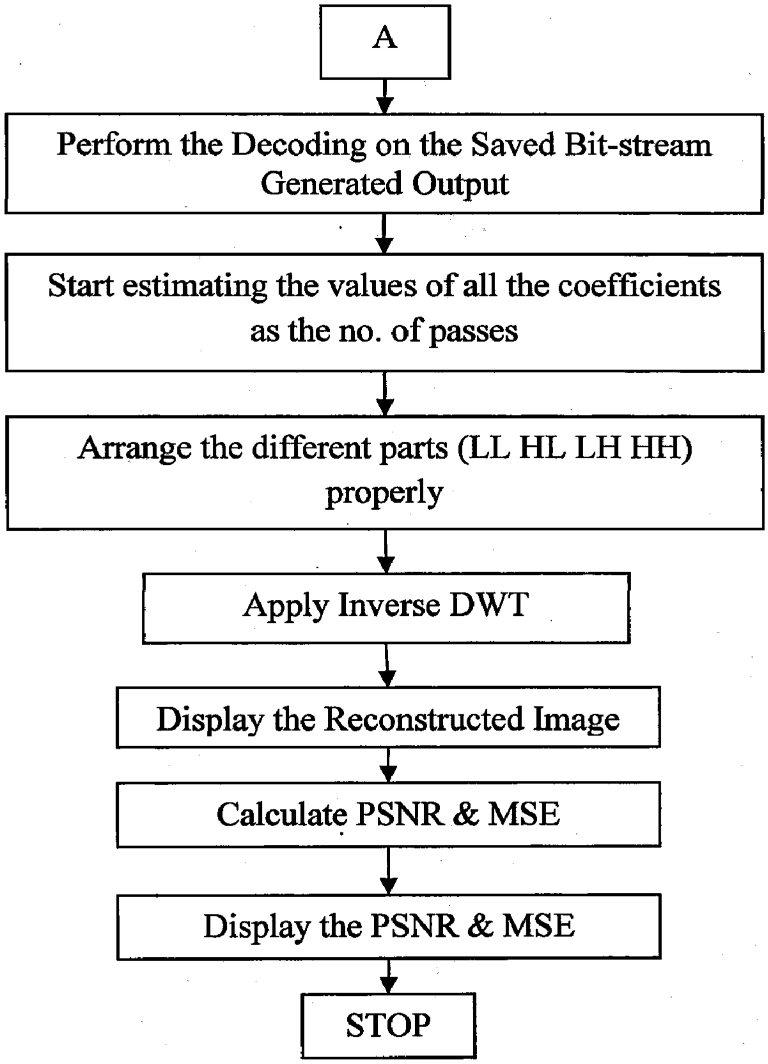
Two linked lists are maintained: LIS – List of Insignificant Sets, and LSP – List of Significant Pixels. The former contains sets of type _ of varying sizes which have not yet been found significant against a threshold ***n*** while the latter obviously contains those pixels which have tested significant against ***n***. Alternatively, we can use an array of smaller lists of type LIS, each containing sets of type ***S*** of a fixed size, instead of using a single large list having sets ***S*** of varying sizes. These smaller lists are ordered by size from smallest single pixel sets first (top) to largest sets last (bottom). This ordering is the functional equivalent of separating the LIS into two lists, an LIP (list of insignificant points) and an LIS (list of insignificant multi-point sets), as done in SPIHT. Use of multiple lists will speed up the encoding/decoding process. [23]

5.2 Features of the coder: [24]

1. It is completely embedded - a single coded bit stream can be used to decode the image at any rate less than or equal to the coded rate, to give the best reconstruction.
2. It employs progressive transmission – source samples are coded in decreasing order of their information content.
3. It has low computational complexity – the algorithm is very simple, consisting mainly of comparisons, and does not require any complex computation.
4. It has low dynamic memory requirements – at any given time during the coding process, only one connected region (lying completely within a Subband) is processed. Once this region is processed, the next region is then considered for processing.
5. It has fast encoding/decoding - this is due to the low-complexity nature of the algorithm.
6. It has efficient performance - its efficiency is comparable to the other low-complexity algorithms available today.

5.3 SBHP FLOWCHART:





5.4 SBHP ALGORITHM [23]

1) **Initialization**

   a) Partition image transform $\boldsymbol{x}$ into two sets: $\boldsymbol{S}$ = root, and $\boldsymbol{I} = \boldsymbol{x} - \boldsymbol{S}$.

   b) . output $n_{\max} = \left\lfloor \log_2 \left(\max_{(i,j)\in x} \left\{ |c_{i,j}| \right\} \right) \right\rfloor$

   c) Add $\boldsymbol{S}$ to LIS and set LSP = ϕ

2) **Sorting Pass**

   a) In increasing order of size $|S|$ of sets (smaller sets first) For each set $\boldsymbol{S} \in$ LIS do ***Process(S)***

   b) if $\boldsymbol{I} \neq \phi$, ***Process( )***

3) **Refinement Pass**

For each (i , j) ∈ LSP, except those included in the last sorting pass, output the n-th MSB of $|c_{i,j}|$

4) **Quantization Step**

Decrement ***n*** by 1, and go to step 2

Procedure for Process S ( )

1) Output $\tau_n(S)$

2) If $\tau_n(S)=1$

a) If ***S*** is a pixel, then output sign of ***S*** and add ***S*** to LSP

b) else ***CodeS(S)***

if ***S*** ∈ LIS, then remove ***S*** from LIS

3)else

If ***S*** ∉ LIS, then add ***S*** to LIS

4) return

Procedure for *Code S ( )*

1) Partition ***S*** into four equal subsets $O(S)$

2) for each set $S_i \in 0(S)(i=1,2,3)$

- output $\tau_n(S_i)$
- if $\tau_n(S_i)=1$

– if S_i is a pixel, output its sign and add S_i to LSP

– else ***CodeS(*** S_i ***)***

- else

– add S_i to LIS

3) return

Procedure for *Process I( )*

1) output $\tau_n(I)$

2) if $\tau_n(I)=1$

- ***CodeI( )***

3) return

Procedure for *Code I( )*

1) Partition ***I*** into four sets—three S_i and one ***I***.

2) for each of the three sets S_i $(i = 1,2,3)$

- ***Process S(*** S_i ***)***

3) ***Process I ( )***

4) return

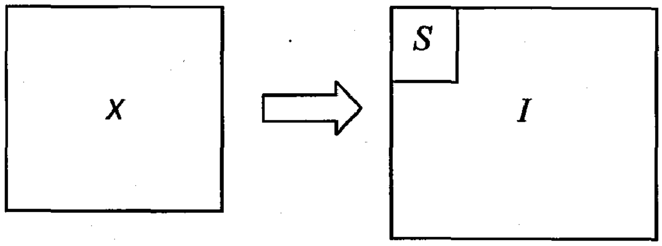


Fig 5.1 Partitioning of image x into sets S and I [23]

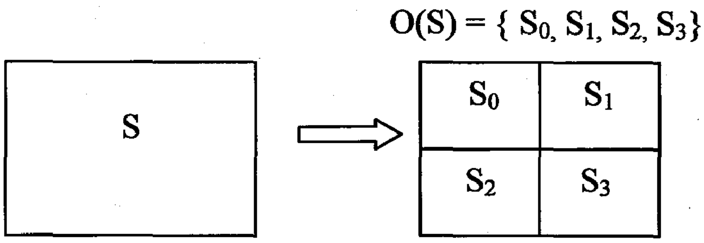


Fig 5.2 Partitioning of set ***S*** [23]

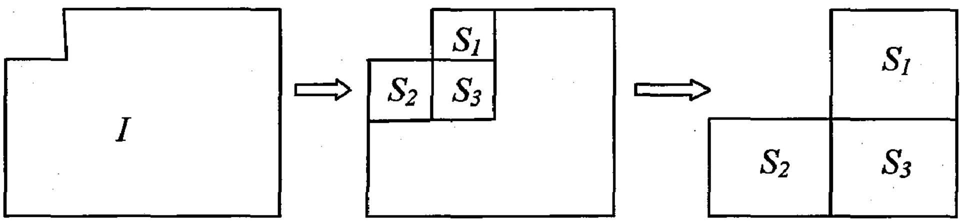


Fig 5.3 Partitioning of set *I* [23]

The algorithm starts by partitioning the image into two sets: set ***S*** which is the root of the pyramid, and set ***I*** which is everything that is left of the image after taking out the root. To start the algorithm, set ***S*** is added to the LIS. A note of the maximum threshold $n_{\max}$ is maintained such that $c_{i,j}$ is insignificant with respect to $n_{\max}+1, \forall_{c_{i,j}} \in x$, but at least one $c_{i,j} \in x$ is significant against the threshold $n_{\max}$.

5.4.1 Quadtree Partitioning:

Set ***S*** in LIS is processed by testing it for significance against the threshold $n = n_{\max}$ (*function ProcessS()*). If not significant, it stays in the LIS. If ***S*** is significant, it is quadrisected, i.e., partitioned into four subsets ***O(S),*** each having size approximately one-fourth the size of the parent set ***S*** (*function CodeS()*). Fig gives an illustration of this partitioning process. In the following procedure *CodeS()*, each of these offspring sets ***O(S)*** is tested for significance for the same ***n*** and, if significant, is quadrisected once more. If not significant, it is added to the LIS.

Each significant subset is, in turn, treated as a set of type ***S*** and processed recursively, via *ProcessS()* and *CodeS()*, until pixel-level is reached where the pixels that are significant in the original set ***S*** are located and thereby coded. The pixels/sets that are found insignificant during this selection process are added to LIS to be tested later against the next lower threshold.

The binary result of every significance test is sent to the code bitstream. Whenever a set ***S*** of size greater than one is significant, tests of four offspring sets follow, so that the binary significance decision paths map onto a quadtree. The motivation for this so-called quadtree partitioning of such sets achieves two goals: (1) to quickly identify the areas of high energy (magnitude) in the set ***S*** and code them first; and (2) to locate structured groups of coefficients that are below a decreasing sequence of magnitude thresholds, so as to limit the number of bits needed for their representation. [23]

5.4.2 Octave Band Partitioning:

At this stage of the algorithm, all current sets of type ***S*** have been tested against ***n***. The set ***I*** is processed next, by testing it against the same threshold ***n*** (*function ProcessI()*). If it is found to be significant, it is partitioned by yet another partitioning scheme—the octave band partitioning. Fig. gives an illustration of this partitioning scheme. Set is partitioned into four sets—three sets of type ***S*** and one of type ***I*** (*function CodeI()*). The size of each of these three sets ***S*** is the same as that of the chopped portion of ***x***. The new set ***I*** that is formed by this partitioning process is now reduced in size.

The idea behind this partitioning scheme is to exploit the hierarchical pyramidal structure of the Subband decomposition, where it is more likely that energy is concentrated at the top most levels of the pyramid and as one goes down the pyramid, the energy content decreases gradually. If a set ***I*** is significant against some threshold ***n***, it is more likely that the pixels that cause ***I*** to be significant lie in the top left regions of ***I***. These regions are decomposed into sets of type ***S***, and are put next in line for processing.

In this way, regions that are likely to contain significant pixels are grouped into relatively smaller sets and processed first, while regions that are likely to contain insignificant pixels are grouped into a large set. A single bit may be enough to code this large region against the particular threshold. Hence, once the set ***I*** is partitioned by the octave band partitioning method, the three sets ***S*** are processed in the regular image-scanning order, after which the newly formed reduced set ***I*** is process. [23]

5.5 Interim conclusions

In this chapter, I explain the SBHP Algorithm and how it considers energy concentration present in both frequency and spatial locations. I draw the flow chart for SBHP Algorithm. I also gave the features of the SBHP coder.

Chapter 6

COMPARATIVE ANALYSIS

6.1 Results on Liver Ultrasound images:

Image 1:

Table 6.1.JPEG Output for Liver Image1:

coefficients	Mse	Snr	Psnr	cr
1	47	31	72	34
2	42	31	74	22
3	36	32	75	17
4	34	32	76	15
5	31	33	77	13

Table 6.2.JPEG2000 Output for Liver Image1:

loop	Mse	snr	Psnr	cr
6	20	35	81	38
7	6	40	92	26
8	1	47	109	19
9	0	54	126	14
10	0	61	142	12

Table 6.3.SBHP Output for Liver Image1:

loop	Mse	snr	Psnr	cr
7	21	34	81	62
8	7	39	92	37
9	1	46	108	25
10	0	53	123	19
11	0	56	129	15

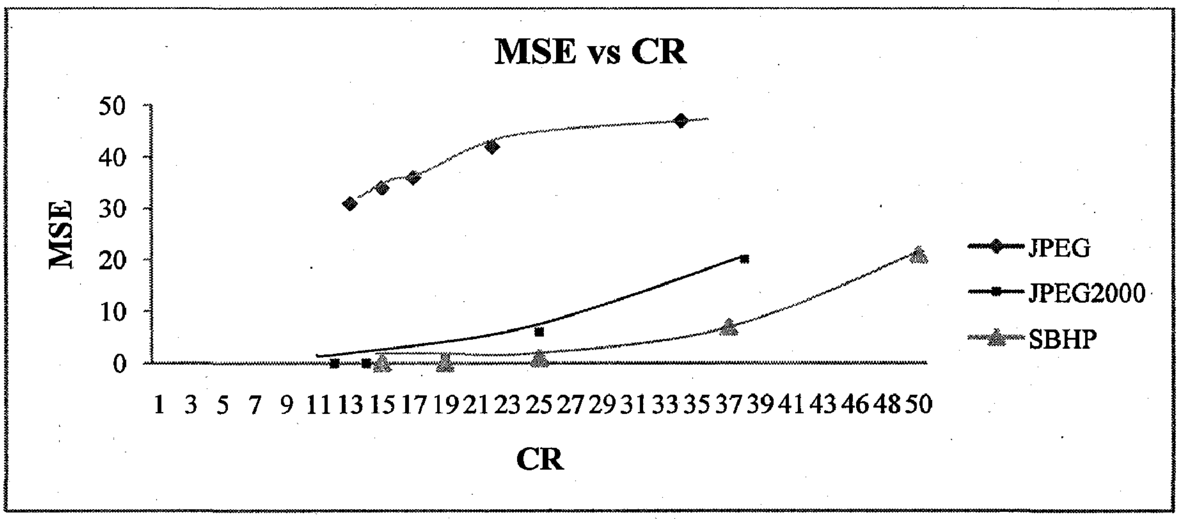


Fig 6.1 A Graph between MSE vs CR

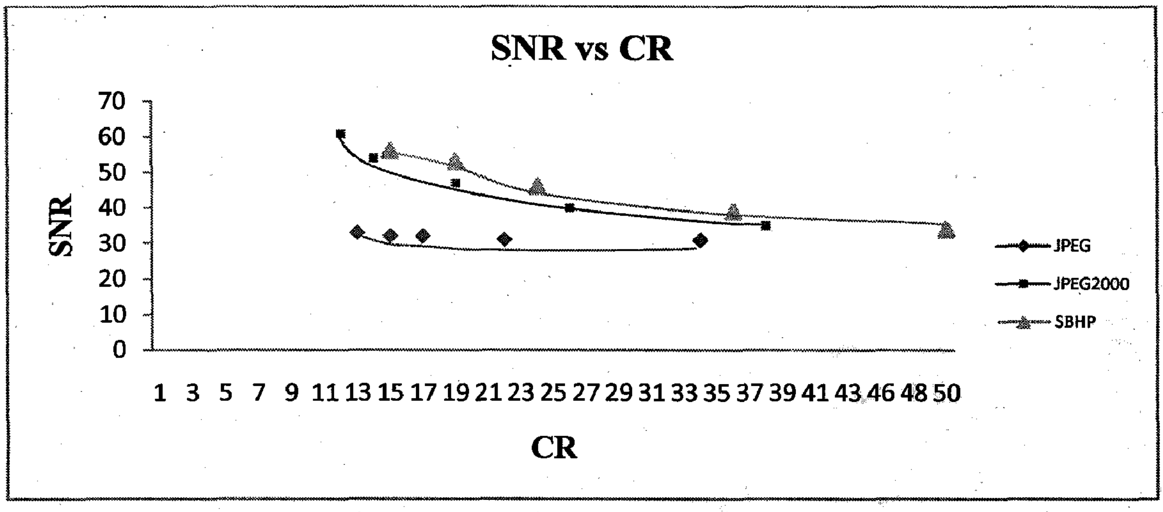


Fig 6.2 A Graph between SNR vs CR

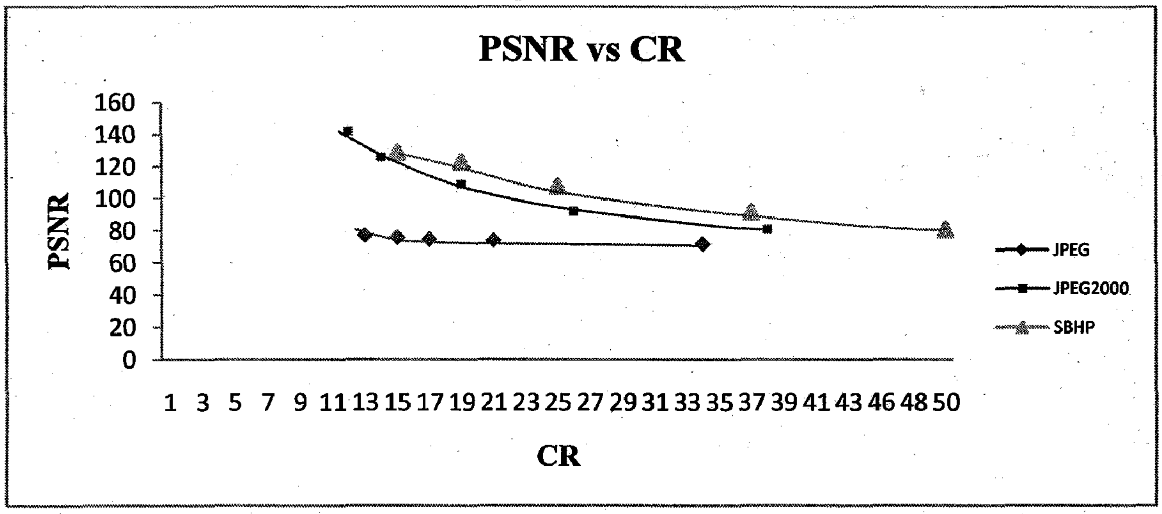


Fig 6.3 A Graph between PSNR vs CR

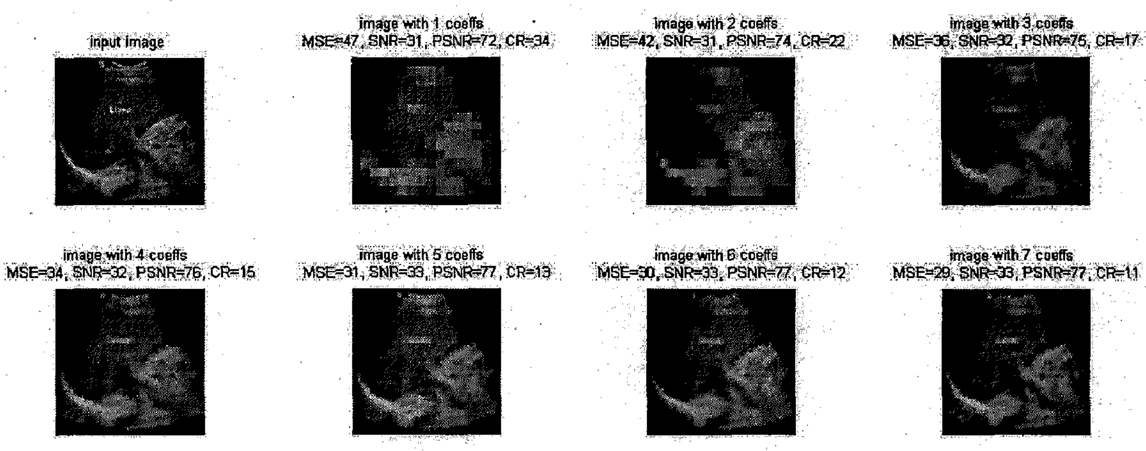


Fig 6.4 Output of JPEG

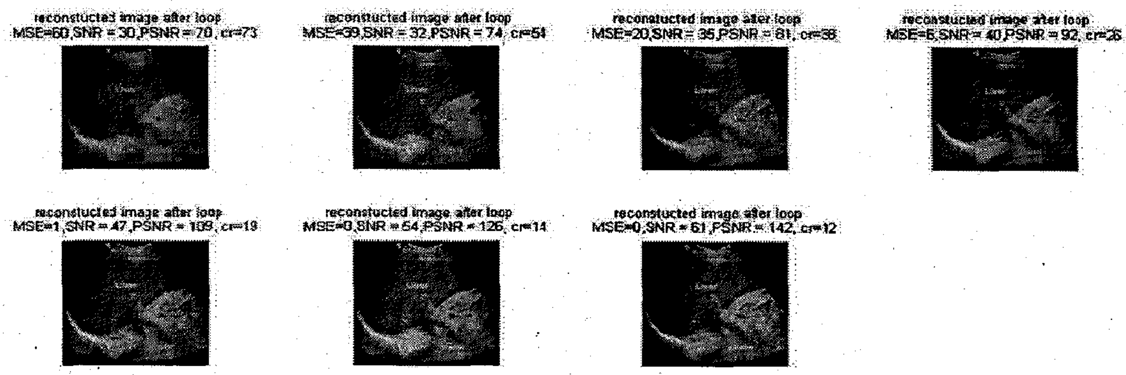


Fig 6.5 Output of JPEG 2000

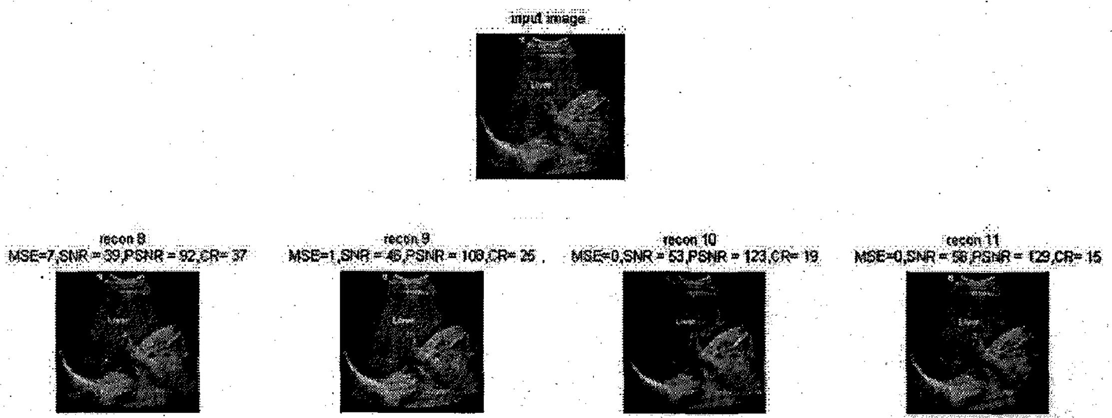


Fig 6.6 Output of SBHP

The three graphs are plotted from the table 6.1 & 6.2& 6.3 gives the following results.

From the graph 6.1, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.2, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.3, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 2:

Table 6.4 JPEG Output for Liver Image2:

coefficients	Mse	snr	Psnr	cr
1	67	29	69	29
2	61	30	70	19
3	51	31	72	15
4	48	31	72	13
5	44	31	73	11

Table 6.5 JPEG2000 Output for Liver Image2:

loop	Mse	snr	Psnr	cr
6	6	40	92	28
7	1	47	110	20
8	0	55	127	19
9	0	62	144	12
10	0	61	142	12

Table 6.6 SBHP Output for Liver Image2:

loop	Mse	snr	Psnr	cr
7	7	39	92	40
8	1	47	108	26
9	0	53	122	19
10	0	56	129	15
11	0	65	154	12

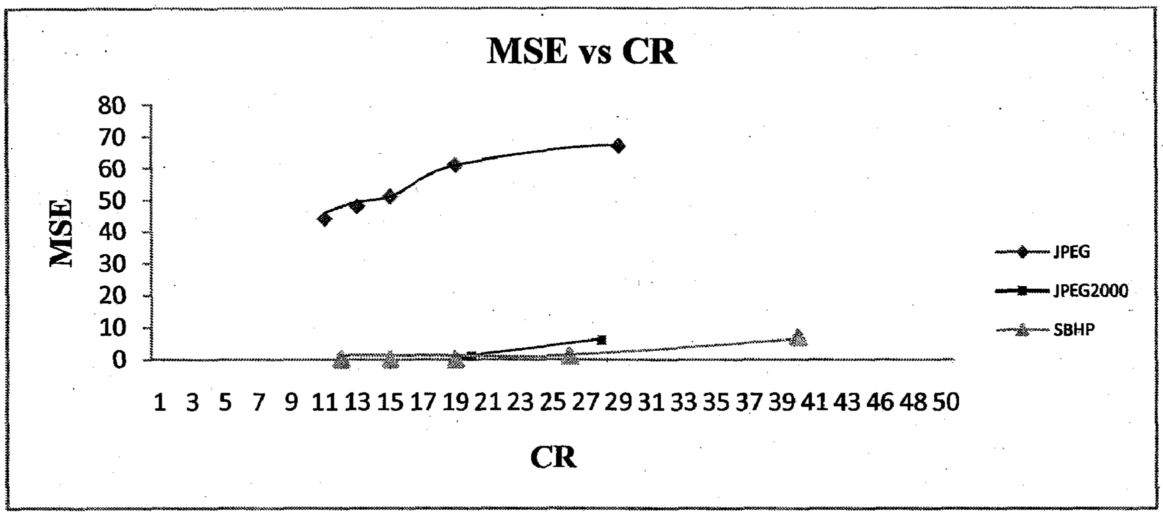


Fig 6.7 A Graph between MSE vs CR

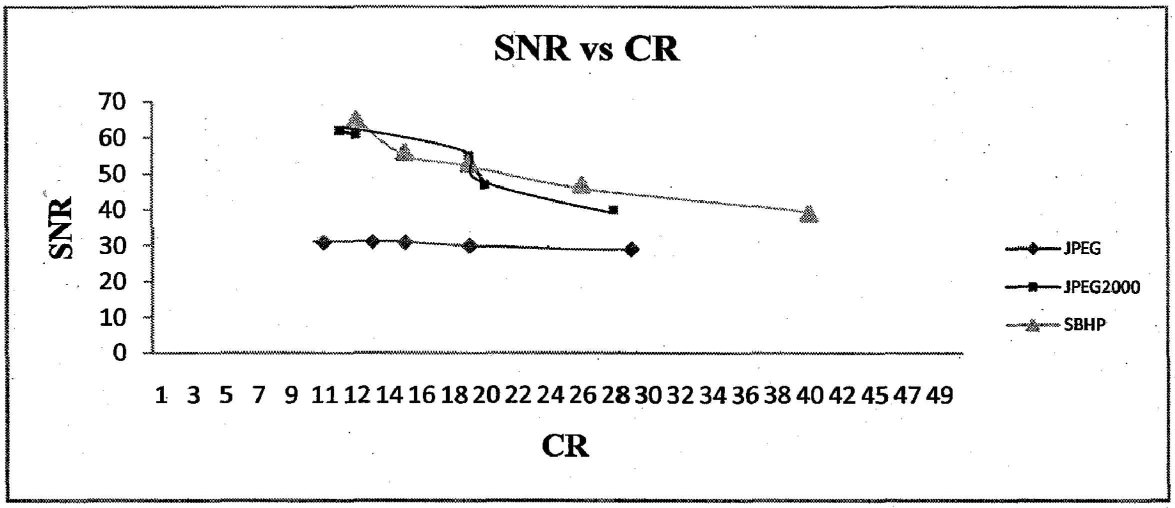


Fig 6.8 A Graph between SNR vs CR

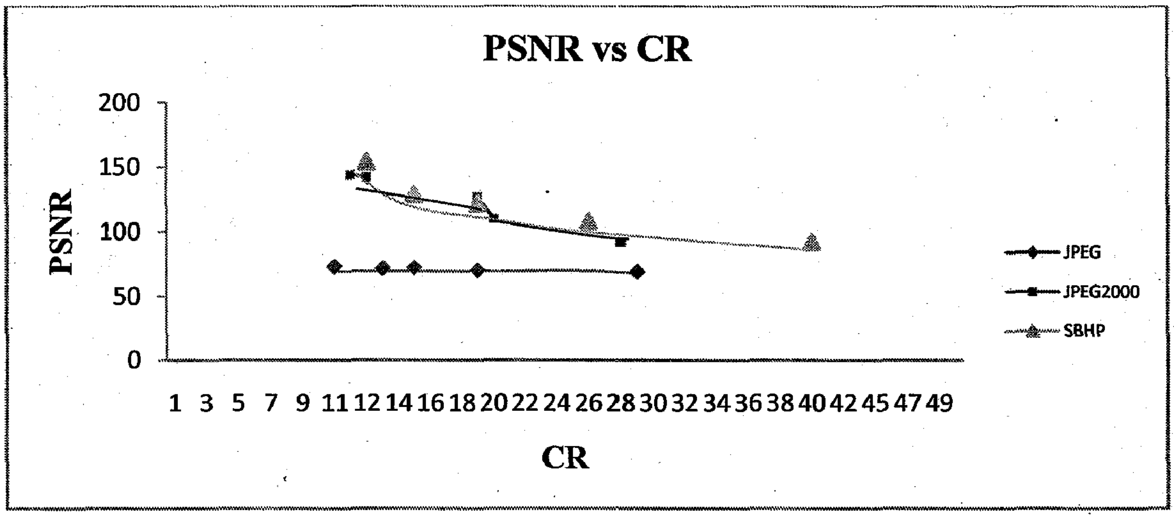


Fig 6.9 A Graph between PSNR vs CR

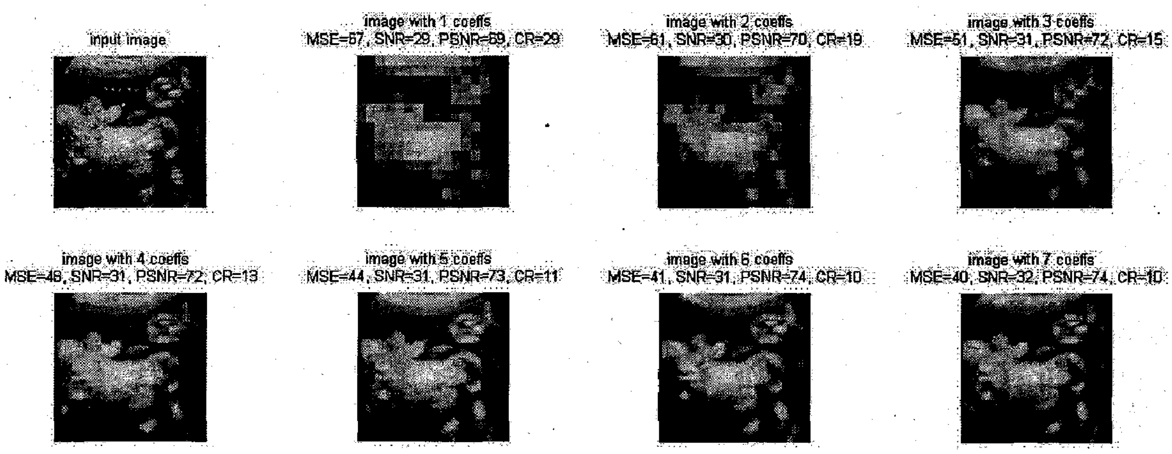


Fig 6.10 Output of JPEG

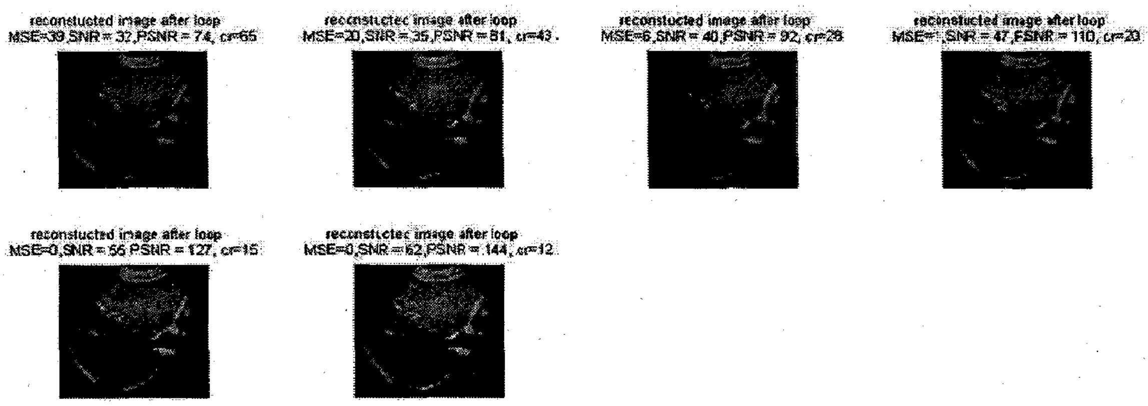


Fig 6.11 Output of JPEG 2000

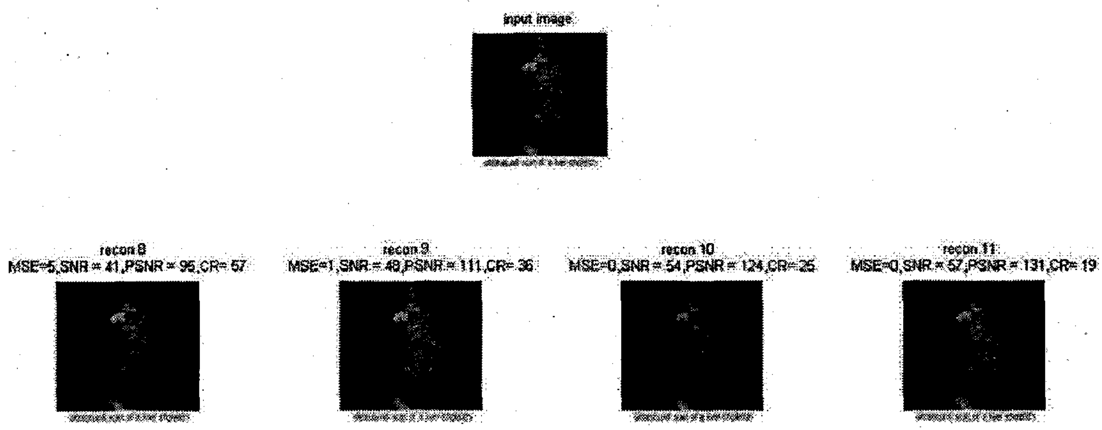


Fig 6.12 Output of SBHP

The three graphs are plotted from the table 6.4 & 6.5& 6.6 gives the following results.

From the graph 6.7, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.
From the graph 6.8, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.
From the graph 6.9, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 3:

Table 6.7.JPEG Output for Liver Image3:

coefficients	Mse	snr	Psnr	cr
1	50	31	72	31
2	46	31	73	19
3	40	32	74	15
4	38	32	74	13
5	35	32	75	12

Table 6.8.JPEG2000 Output for Liver Image3:

loop	Mse	snr	Psnr	cr
6	23	34	80	37
7	7	39	91	24
8	1	46	108	18
9	0	54	125	13
10	0	61	141	11

Table 6.9.SBHP Output for Liver Image3:

loop	Mse	snr	Psnr	cr
7	23	34	79	62
8	7	39	91	34
9	1	46	107	23
10	0	52	120	17
11	0	55	127	14

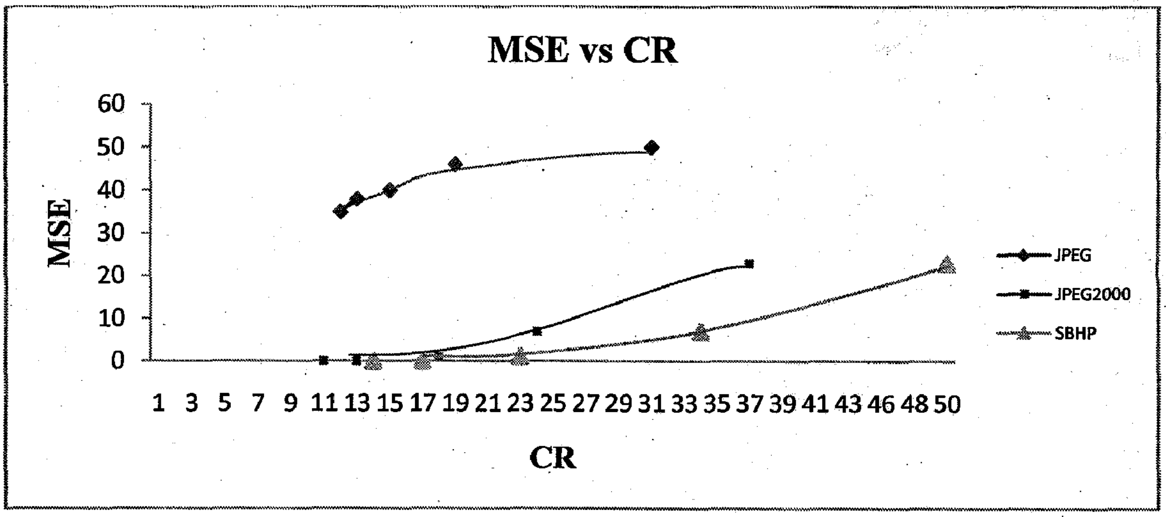


Fig 6.13 A Graph between MSE vs CR

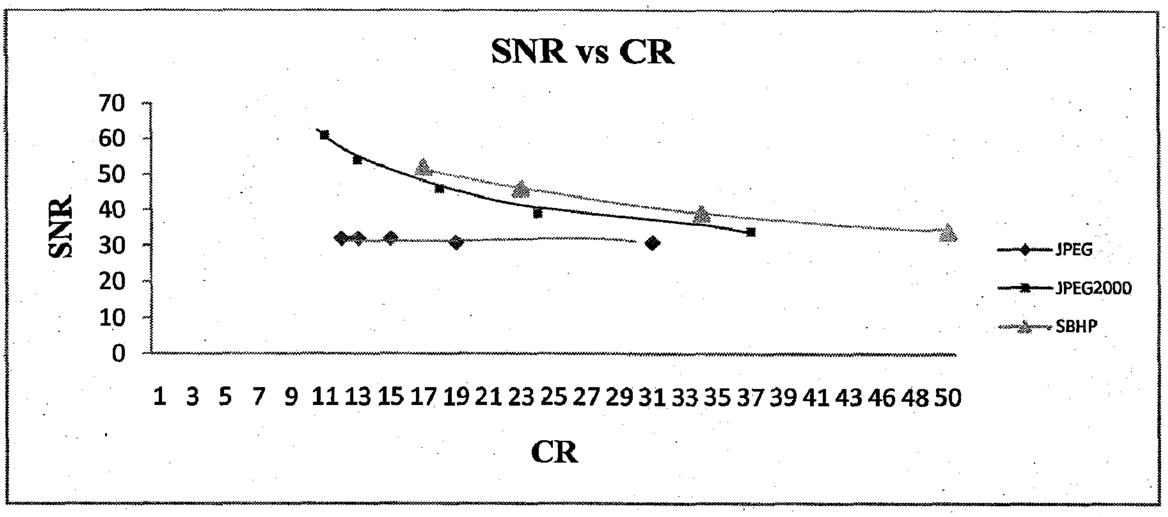


Fig 6.14 A Graph between SNR vs CR

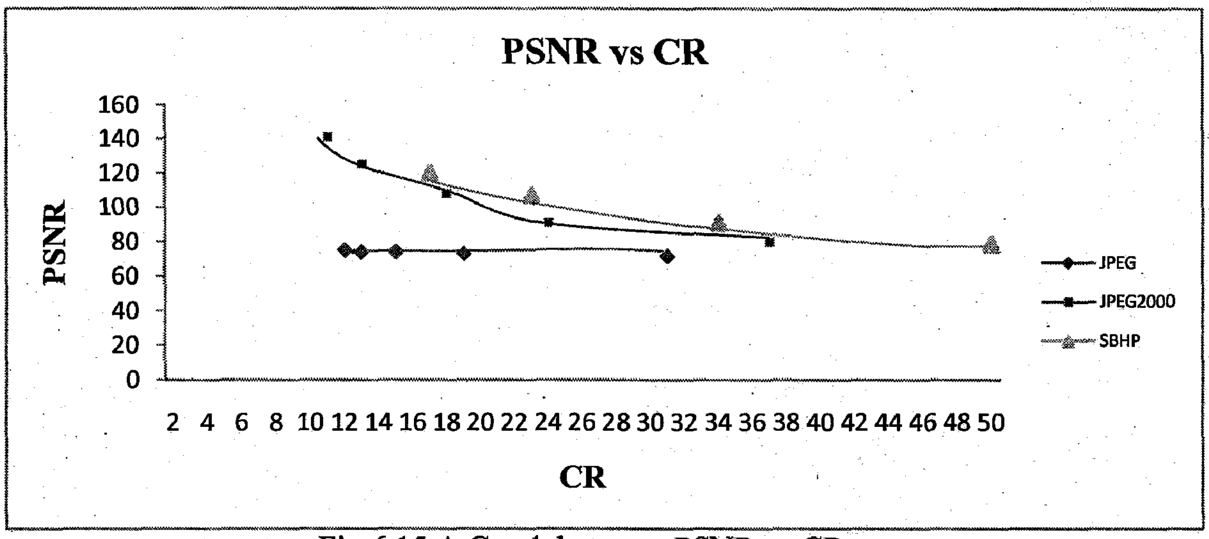


Fig 6.15 A Graph between PSNR vs CR

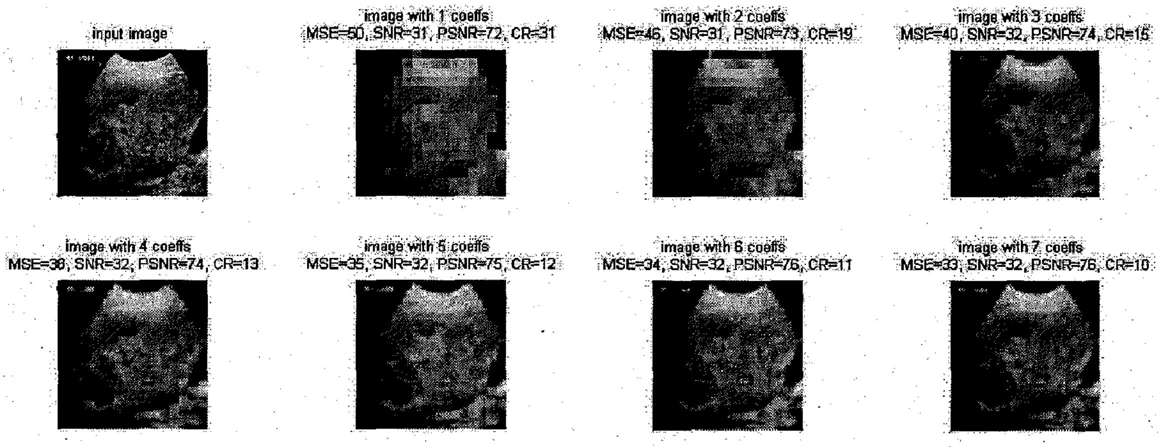


Fig 6.16 Output of JPEG

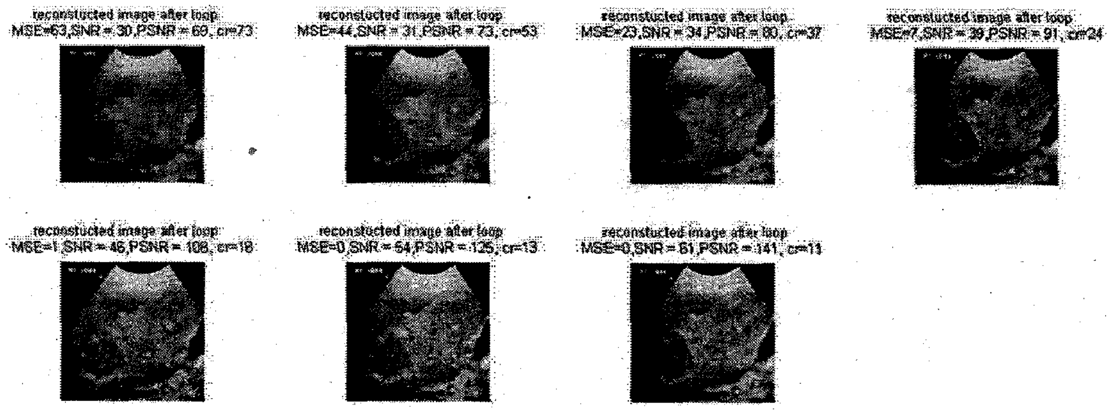


Fig 6.17 Output of JPEG 2000

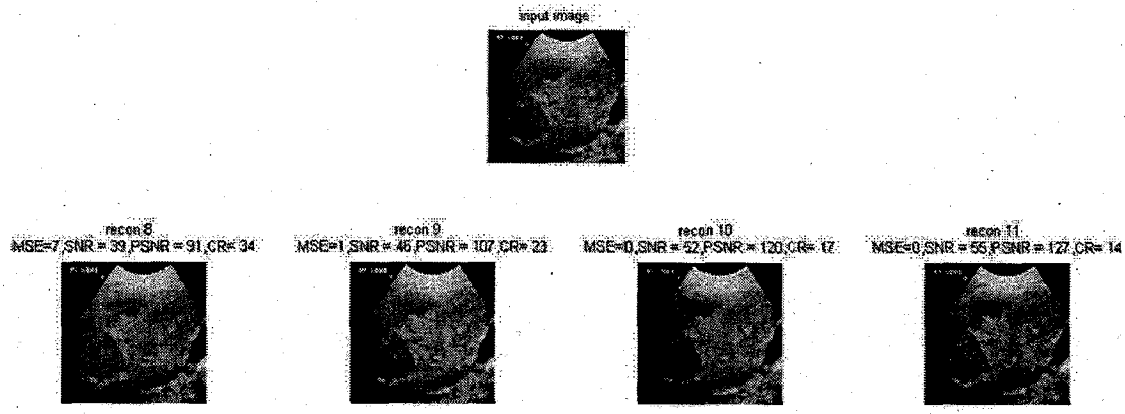


Fig 6.18 Output of SBHP

The three graphs are plotted from the table 6.7 & 6.8 & 6.9 gives the following results.

From the graph 6.13, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.14, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.15, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 4:

Table 6.10.JPEG Output for Liver Image4:

coefficients	Mse	snr	Psnr	cr
1	62	30	70	31
2	58	30	70	20
3	51	31	72	16
4	47	31	72	14
	44	31	73	12

Table 6.11.JPEG2000 Output for Liver Image4:

loop	Mse	snr	Psnr	cr
6	23	34	79	34
7	7	39	91	24
8	1	46	108	17
9	0	54	125	13
10	0	61	141	11

Table 6.12.SBHP Output for Liver Image4:

loop	Mse	snr	Psnr	cr
7	24	34	79	57
8	7	39	91	33
9	2	46	107	22
10	0	52	120	16
11	0	55	127	13

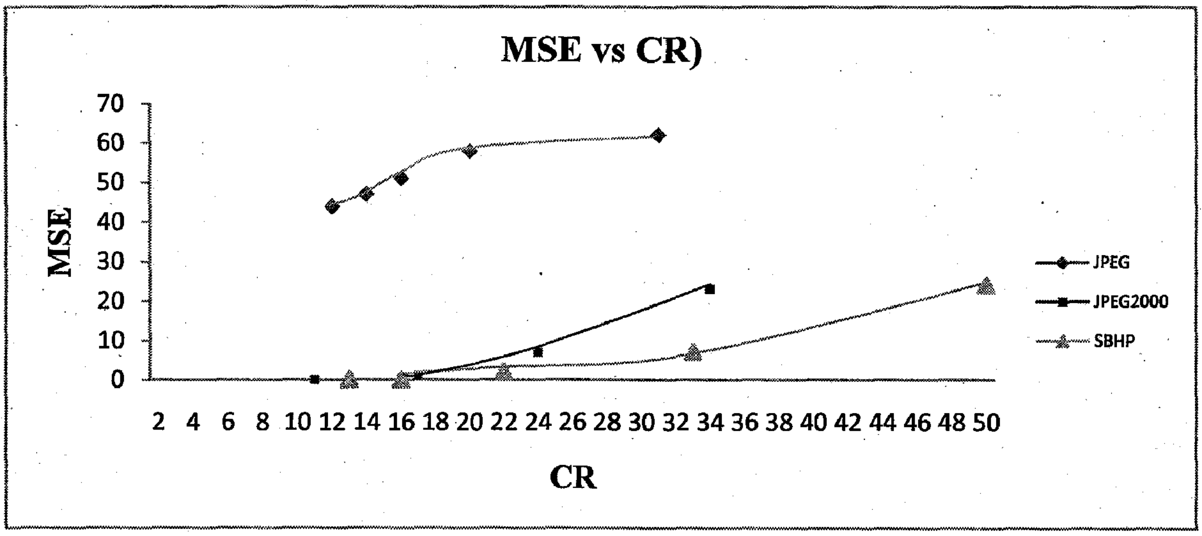


Fig 6.19 A Graph between MSE vs CR

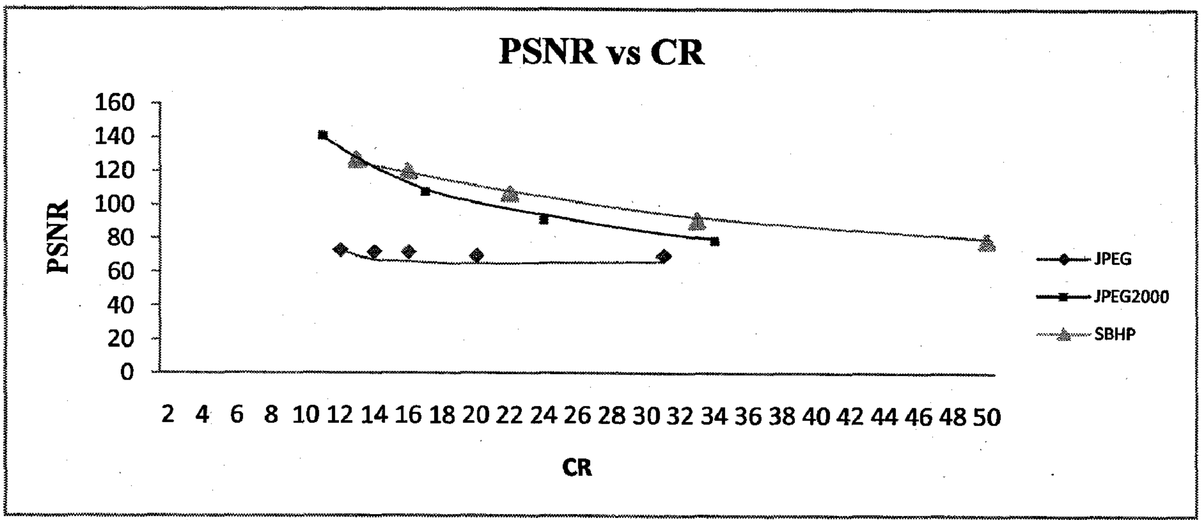


Fig 6.20 A Graph between PSNR vs CR

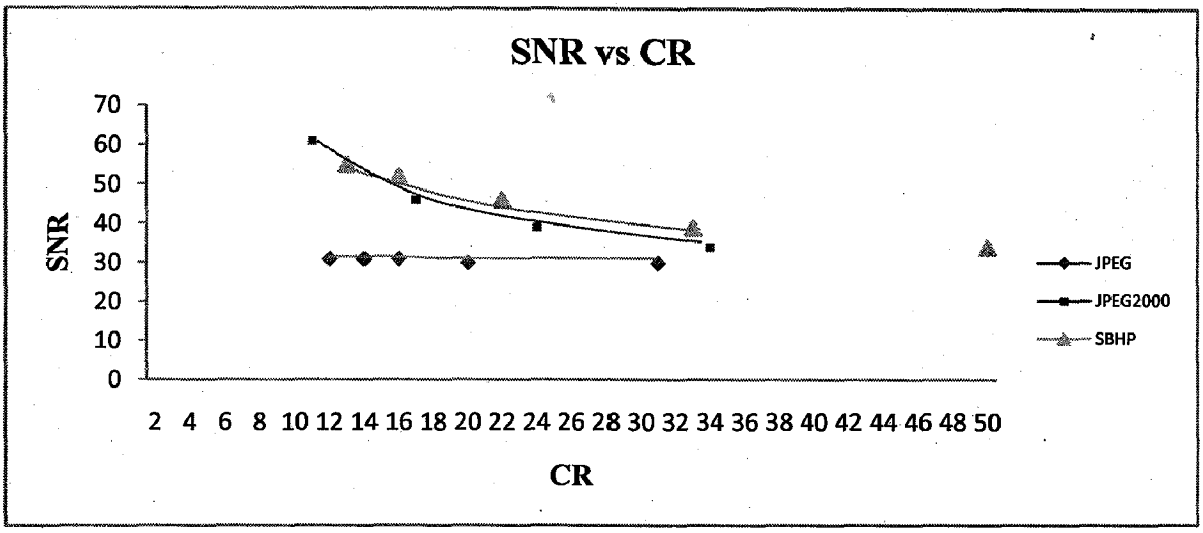


Fig 6.21 A Graph between SNR vs CR

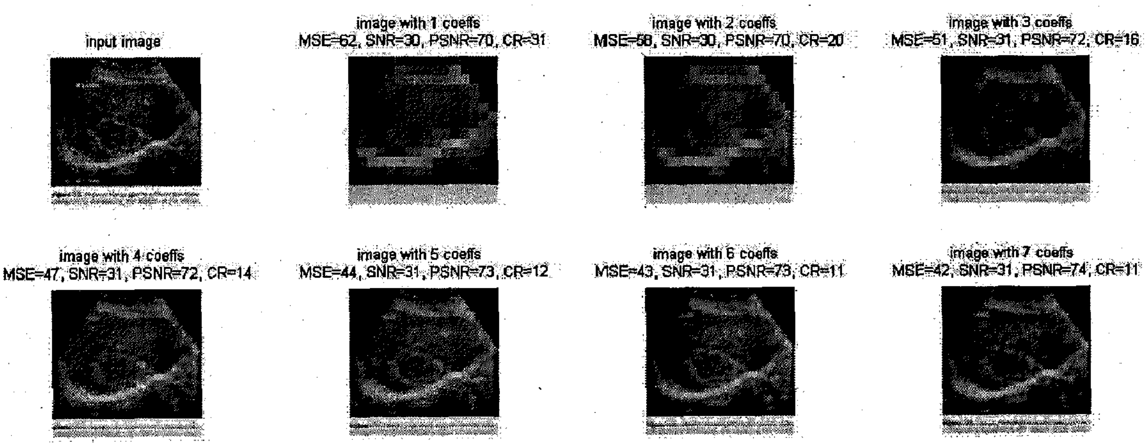


Fig 6.22 Output of JPEG

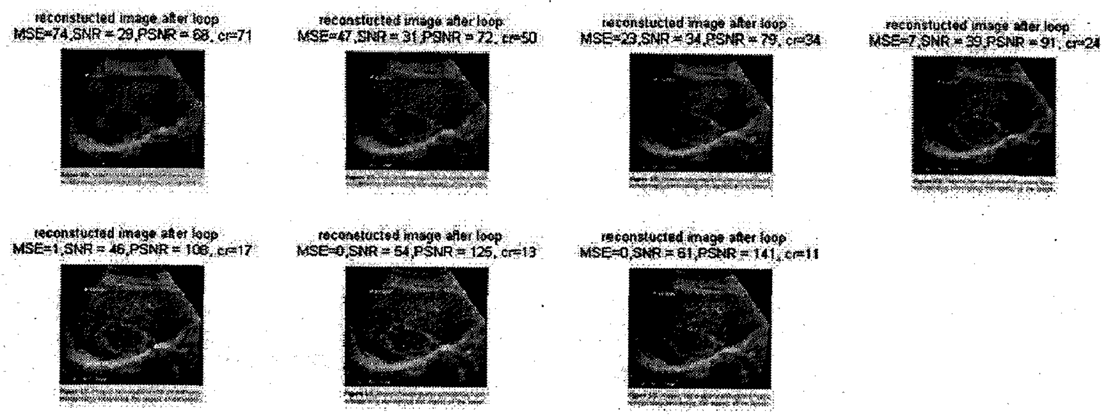


Fig 6.23 Output of JPEG 2000

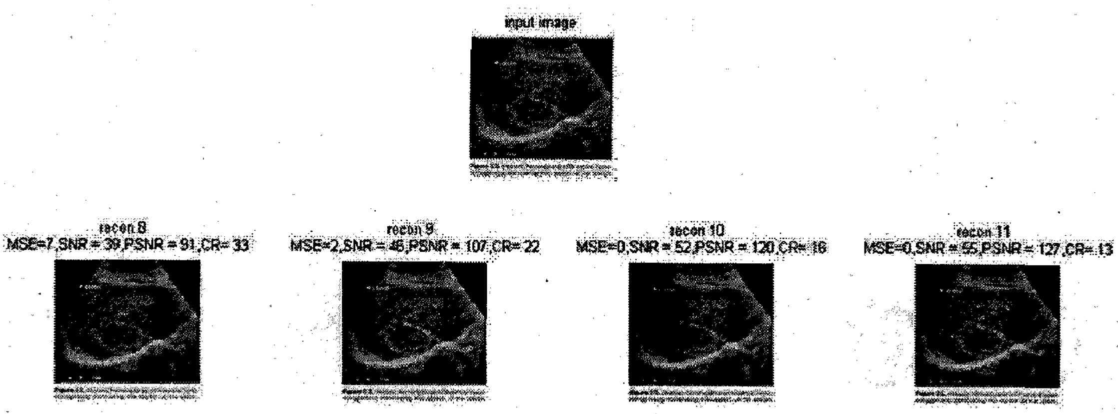


Fig 6.24 Output of SBHP

The three graphs are plotted from the table 6.10 & 6.11& 6.12 gives the following results.

From the graph 6.19, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.21, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.20, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 5:

Table 6.13.JPEG Output for Liver Image5:

coefficients	Mse	snr	Psnr	cr
1	52	30	71	36
2	48	31	72	21
3	43	31	73	17
4	42	31	73	15
5	38	32	74	13

Table 6.14.JPEG2000 Output for Liver Image5:

loop	Mse	snr	Psnr	cr
6	21	34	80	34
7	6	40	93	24
8	1	47	110	18
9	0	55	127	14
10	0	62	143	12

Table 6.15.SBHP Output for Liver Image5:

loop	Mse	snr	Psnr	cr
7	22	34	80	56
8	6	40	92	34
9	1	47	108	24
10	0	53	122	18
11	0	56	129	15

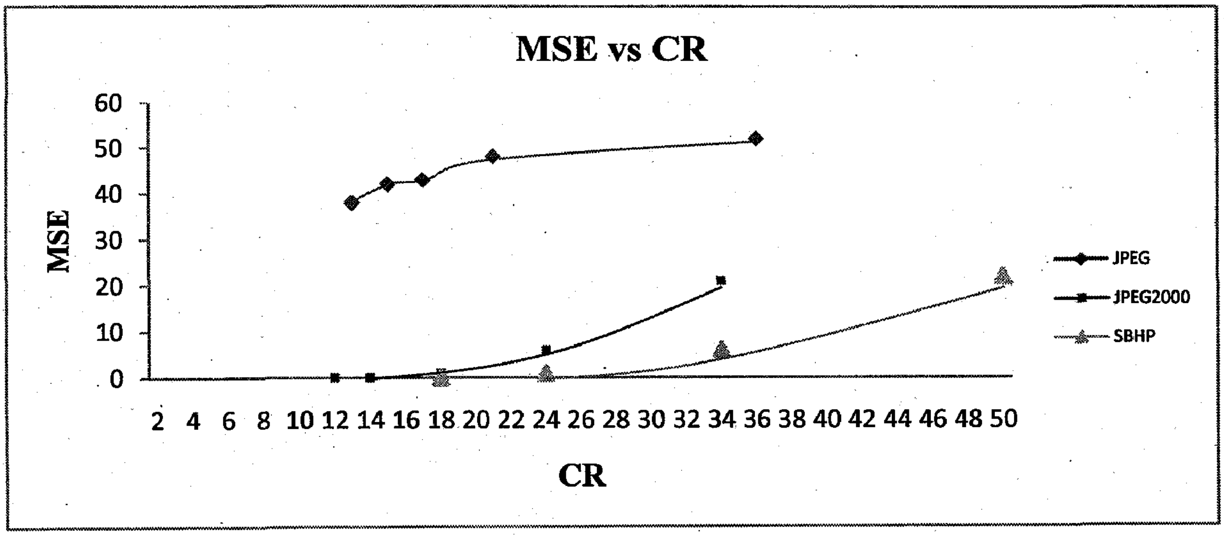


Fig 6.25 A Graph between MSE vs CR

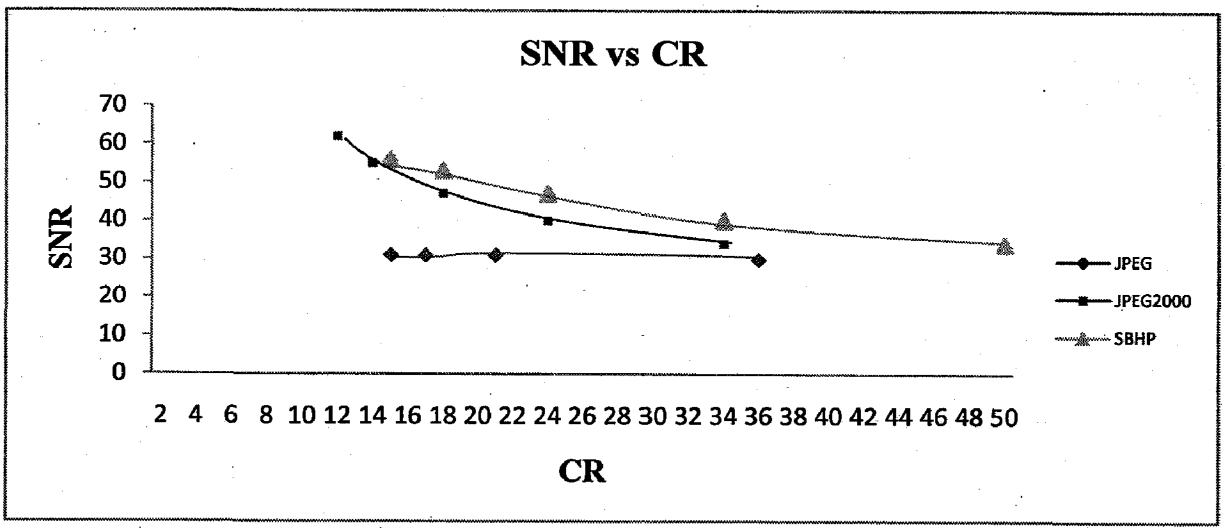


Fig 6.26 A Graph between SNR vs CR

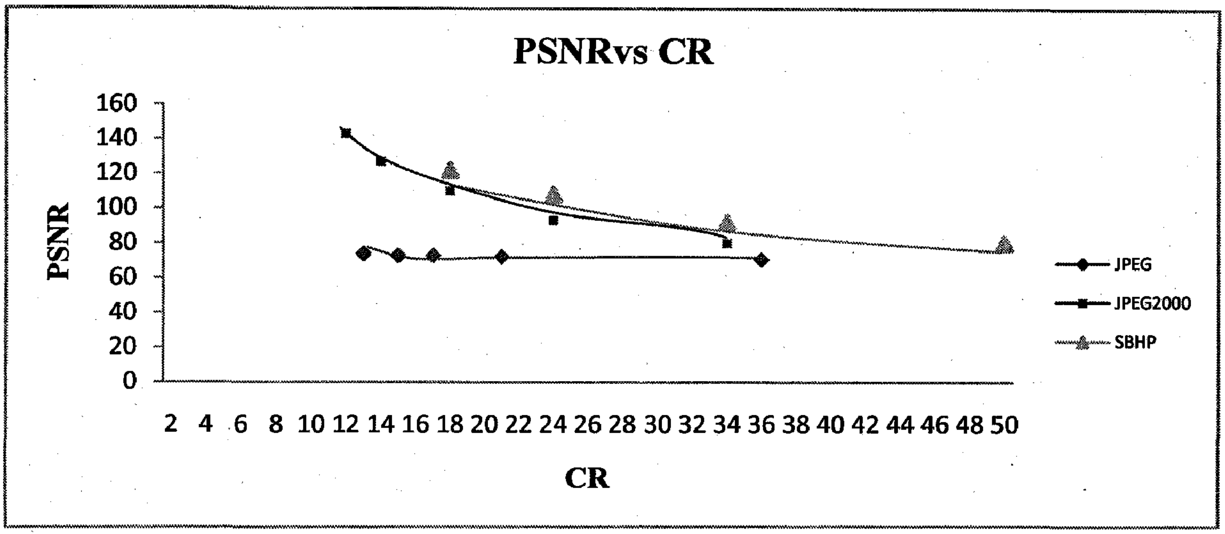


Fig 6.27 A Graph between PSNR vs CR

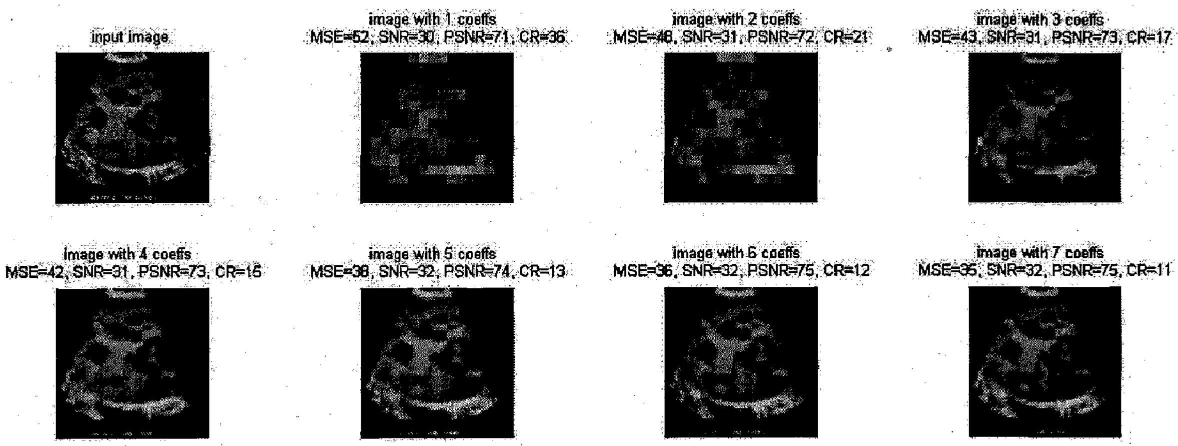


Fig 6.28 Output of JPEG

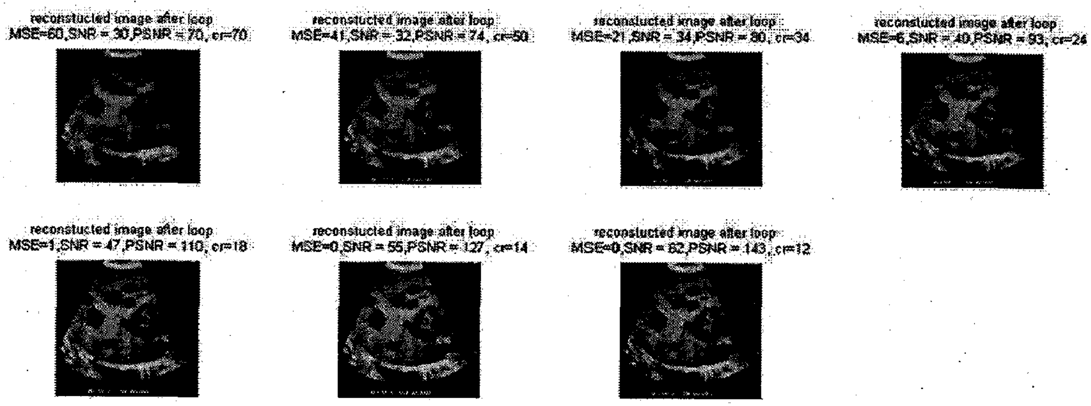


Fig 6.29 Output of JPEG 2000

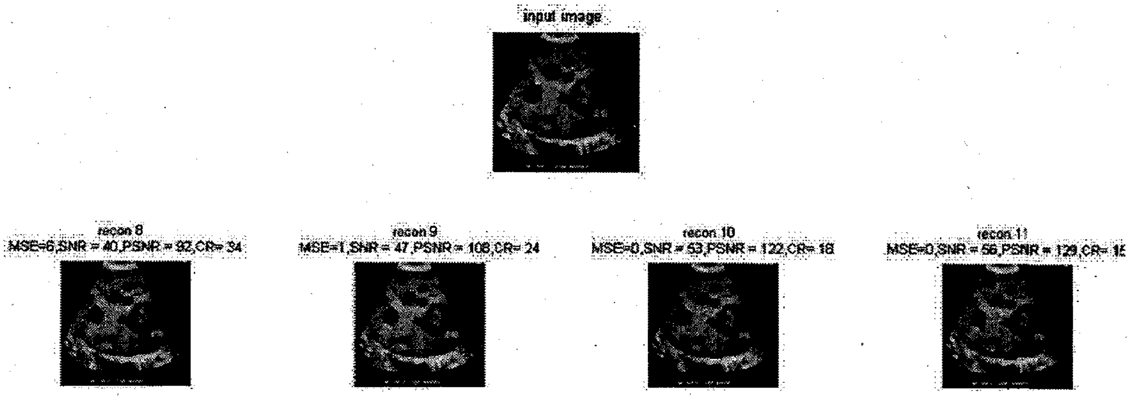


Fig 6.30 Output of SBHP

The three graphs are plotted from the table 6.13 & 6.14& 6.15 gives the following results.

From the graph 6.25, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.26, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.27, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 6:

Table 6.16.JPEG Output for Liver Image6:

coefficients	Mse	snr	Psnr	cr
1	46	31	73	34
2	40	32	74	22
3	33	32	76	17
4	30	33	77	15
5	27	33	78	13

Table 6.17.JPEG2000 Output for Liver Image6:

loop	Mse	snr	Psnr	cr
6	19	35	82	39
7	6	40	93	27
8	1	47	109	19
9	0	54	126	15
10	0	61	142	12

Table 6.18.SBHP Output for Liver Image6:

loop	Mse	snr	Psnr	cr
7	19	35	81	69
8	6	40	92	43
9	1	47	108	27
10	0	52	121	19
11	0	55	128	15

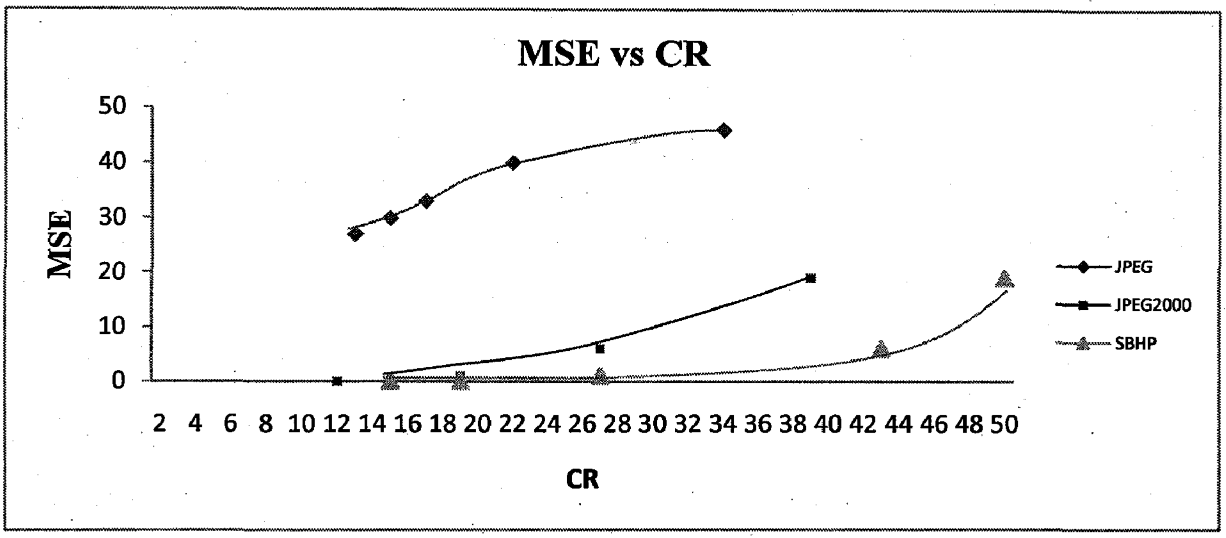


Fig 6.31 A Graph between MSE vs CR

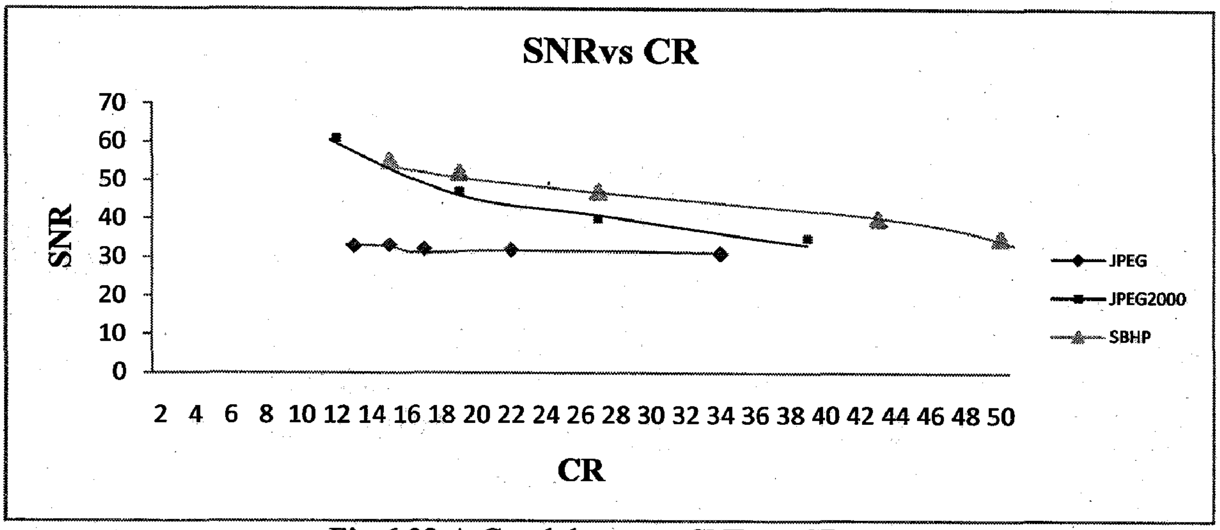


Fig 6.32 A Graph between SNR vs CR

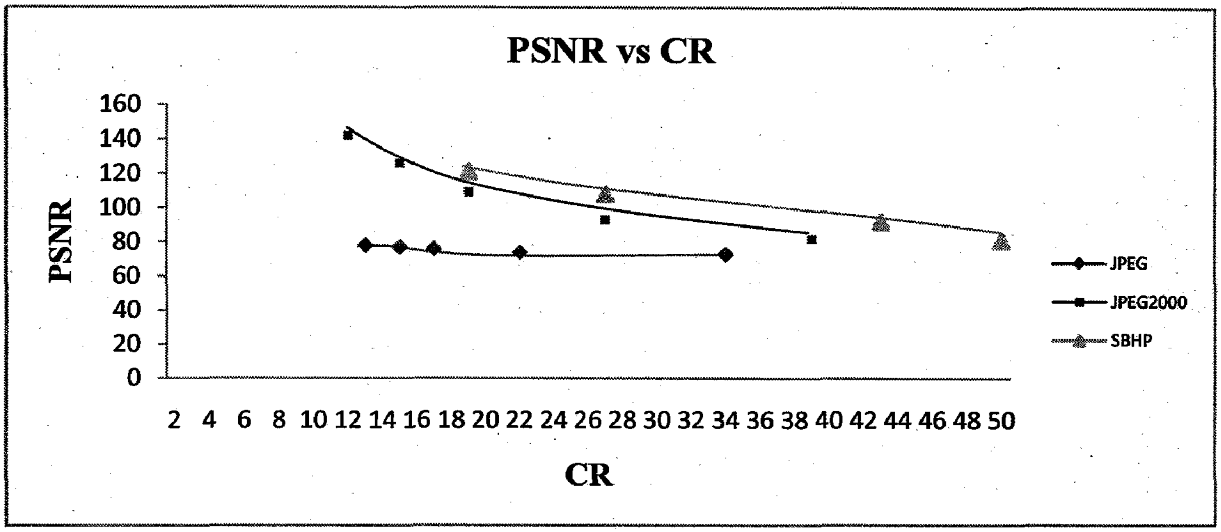


Fig 6.33 A Graph betweenPSNR vs CR

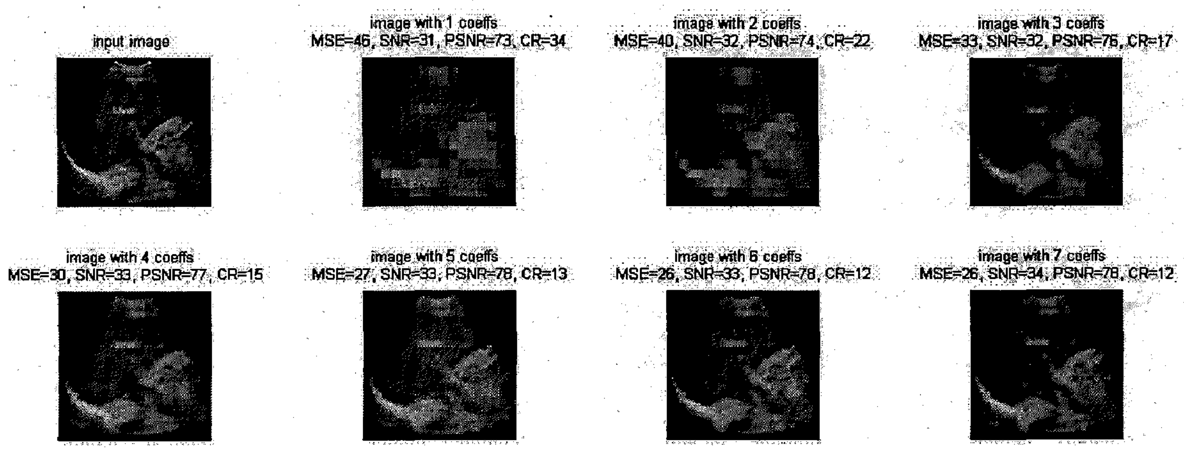


Fig 6.34 Output of JPEG

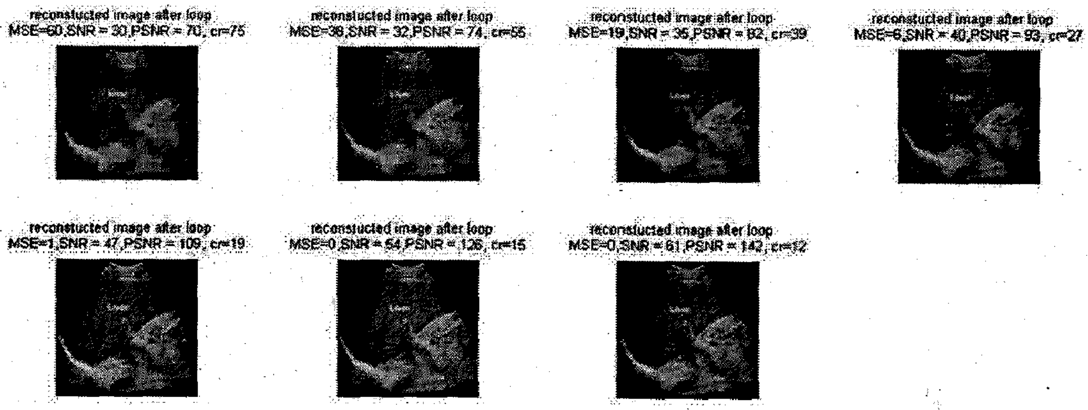


Fig 6.35 Output of JPEG 2000

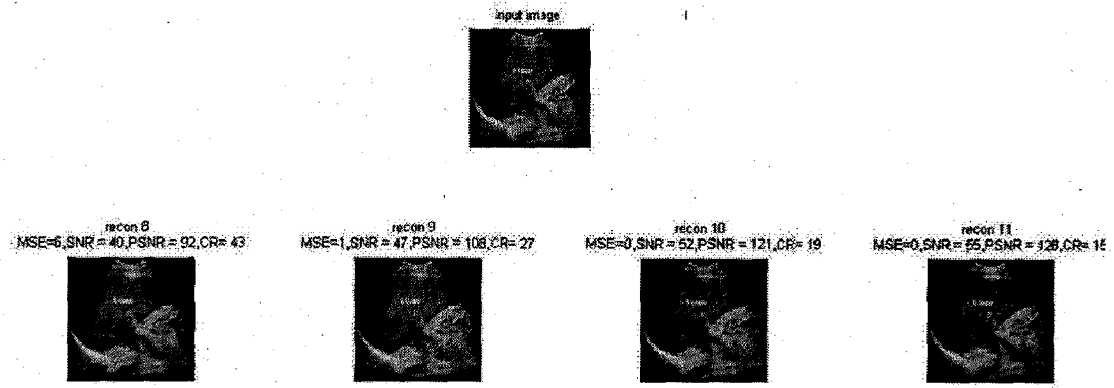


Fig 6.36 Output of SBHP

The three graphs are plotted from the table 6.16 & 6.17& 6.18 gives the following results.

From the graph 6.31, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.32, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.33, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 7:

Table 6.19.JPEG Output for Liver Image7:

coefficients	Mse	snr	Psnr	cr
1	39	32	74	33
2	33	32	76	21
3	26	34	78	17
4	22	34	80	15
5	20	35	81	13

Table 6.20.JPEG2000 Output for Liver Image7:

loop	Mse	snr	Psnr	cr
6	17	35	82	44
7	6	40	93	31
8	1	47	109	22
9	0	53	124	16
10	0	60	140	12

Table 6.21.SBHP Output for Liver Image7:

loop	Mse	snr	Psnr	cr
7	17	35	82	96
8	6	40	93	49
9	1	46	108	32
10	0	52	120	21
11	0	55	127	16

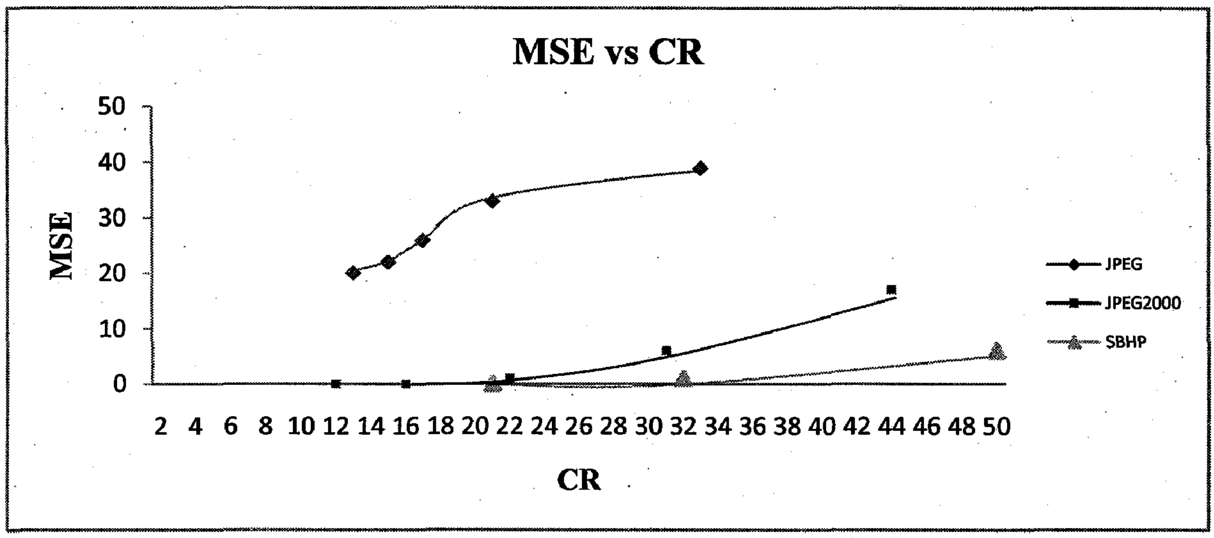


Fig 6.37 A Graph between MSE vs CR

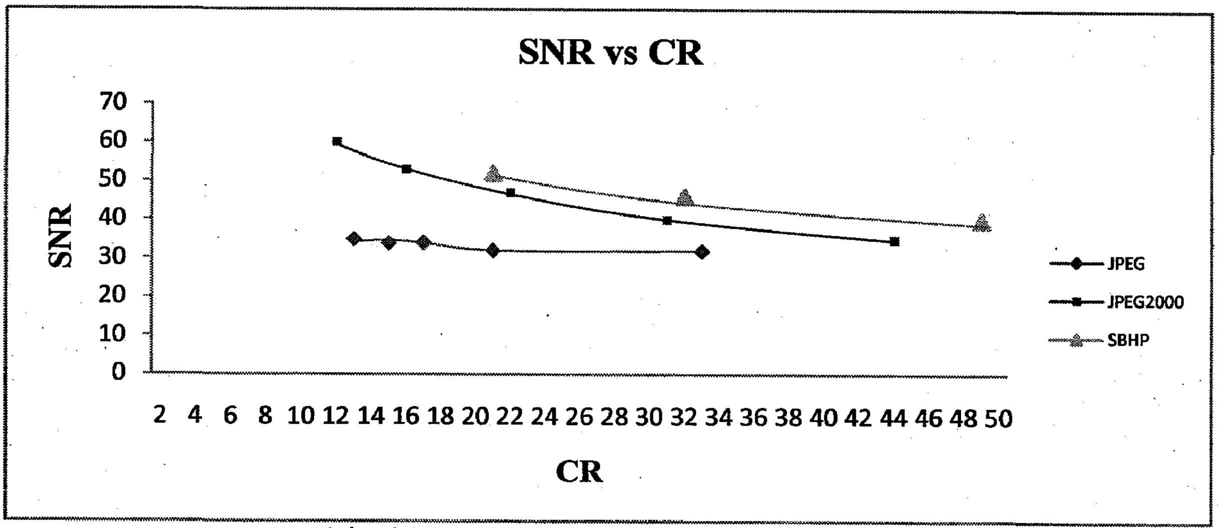


Fig 6.38 A Graph between SNR vs CR

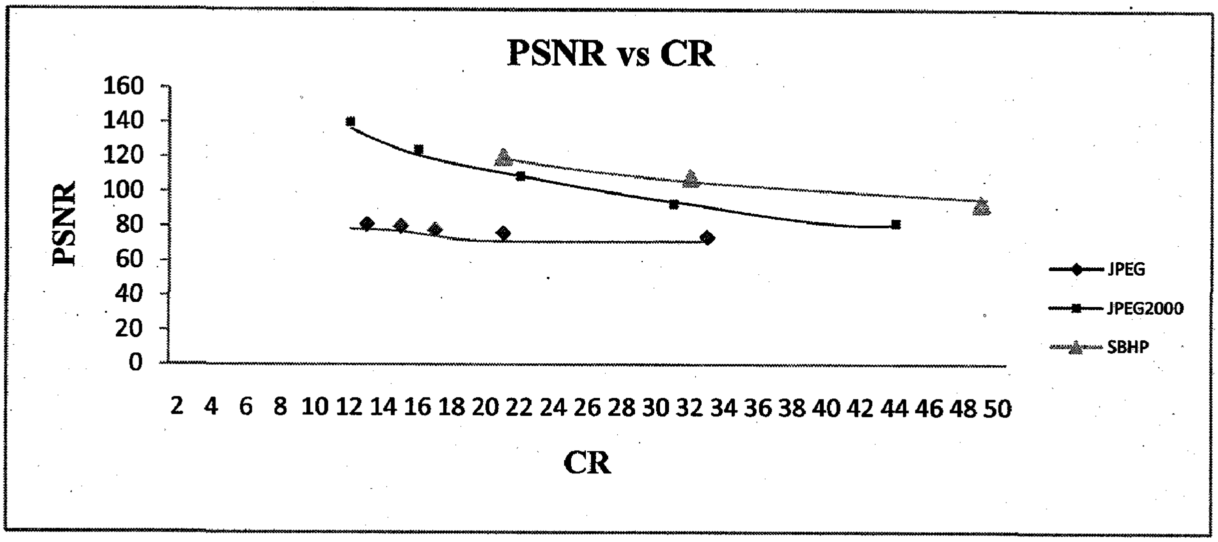


Fig 6.39 A Graph betweenPSNR vs CR

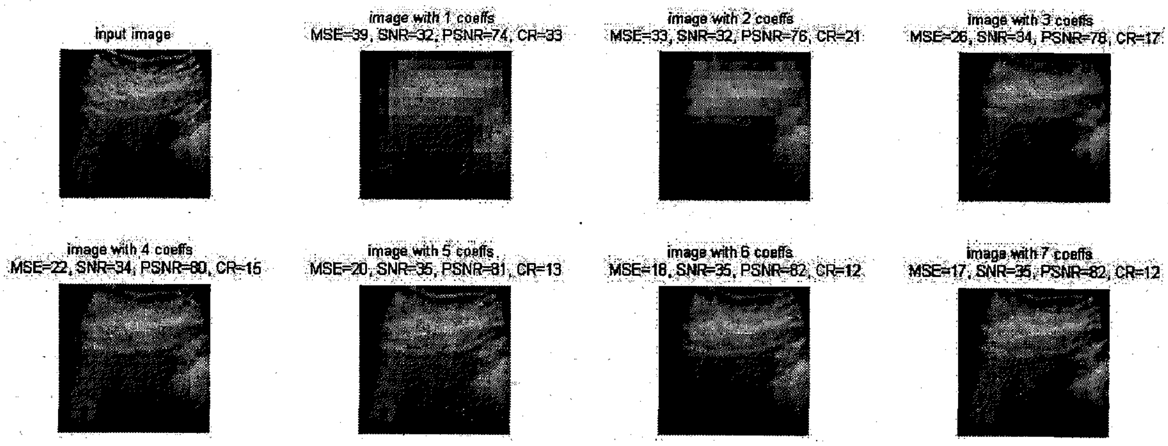


Fig 6.40 Output of JPEG

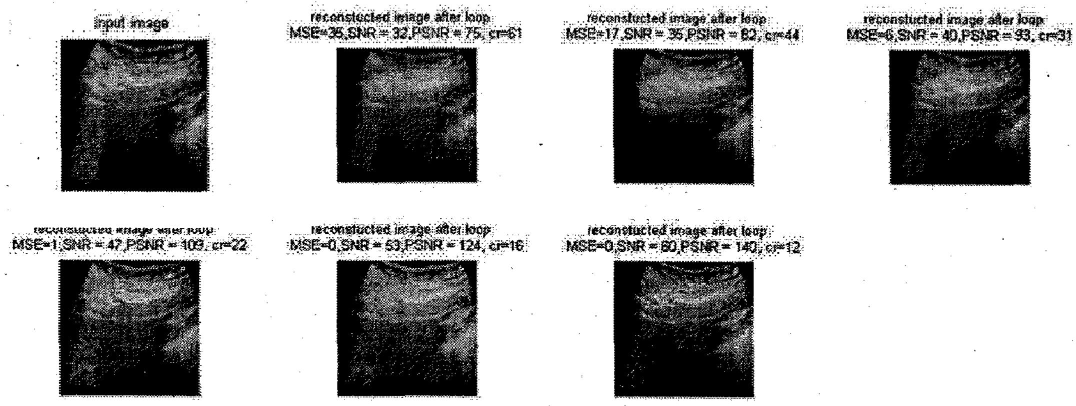


Fig 6.41 Output of JPEG 2000

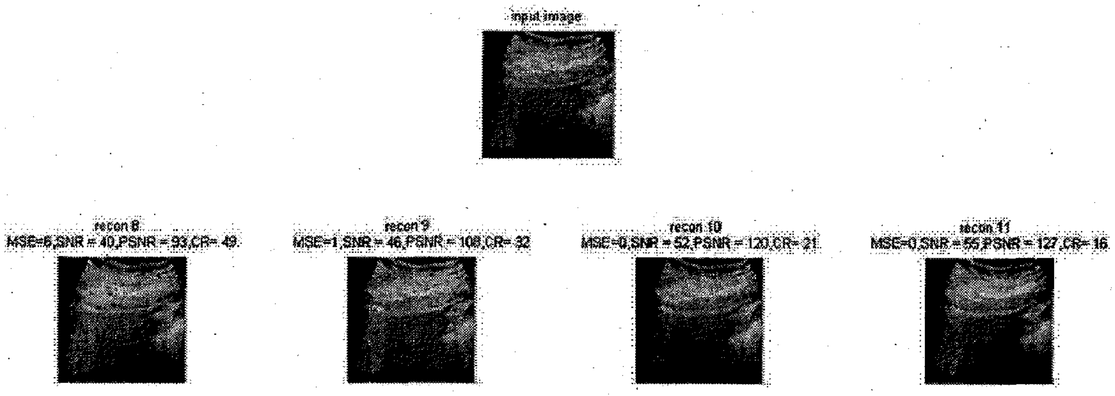


Fig 6.42 Output of SBHP

The three graphs are plotted from the table 6.19 & 6.20& 6.21 gives the following results.

From the graph 6.37, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.38, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.39, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 8:

Table 6.22.JPEG Output for Liver Image8:

coefficients	Mse	snr	Psnr	cr
2	36	32	75	23
3	30	33	77	18
4	28	33	77	16
5	26	33	78	14
6	25	34	79	13

Table 6.23.JPEG2000 Output for Liver Image8:

loop	Mse	snr	Psnr	cr
6	19	35	81	40
7	6	40	93	28
8	1	47	109	20
9	0	55	127	15
10	0	61	142	12

Table 6.24.SBHP Output for Liver Image8:

loop	Mse	snr	Psnr	cr
7	19	35	81	71
8	6	40	93	43
9	1	47	108	28
10	0	52	122	20
11	0	55	129	16

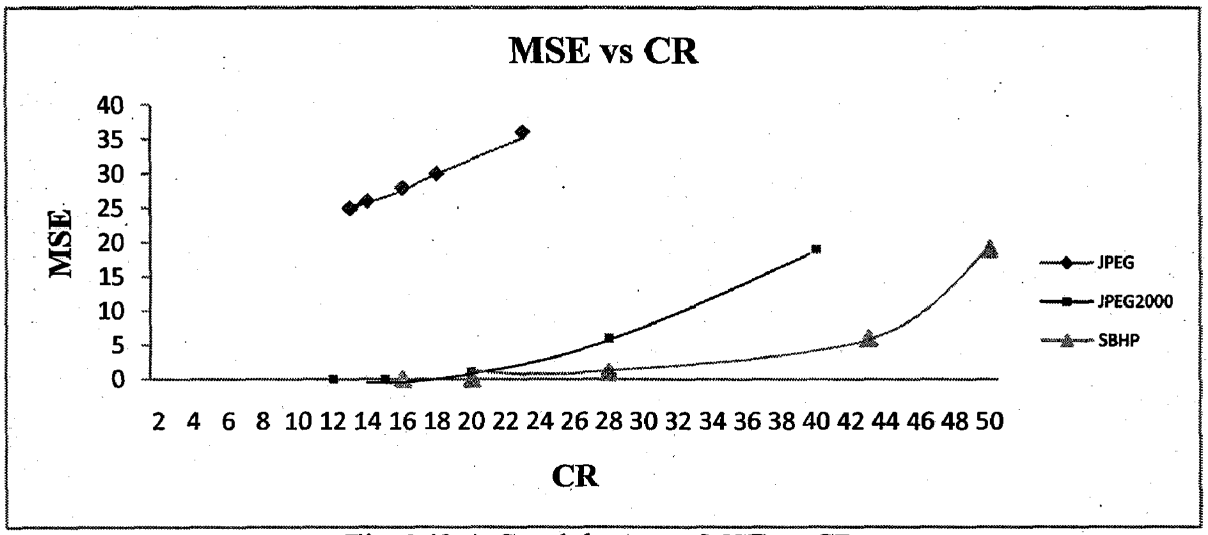


Fig 6.43 A Graph between MSE vs CR

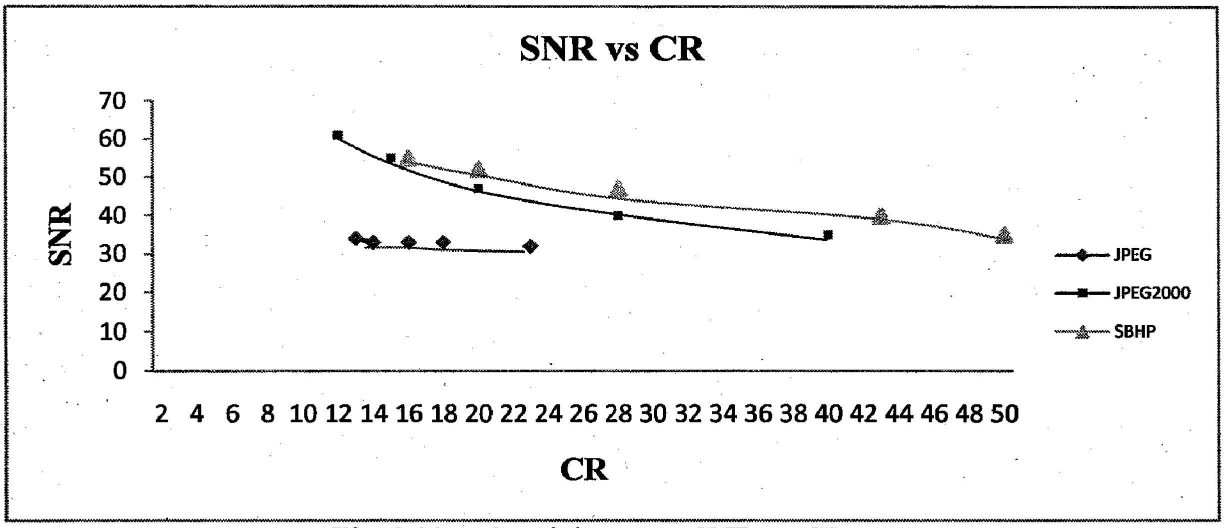


Fig 6.44 A Graph between SNR vs CR

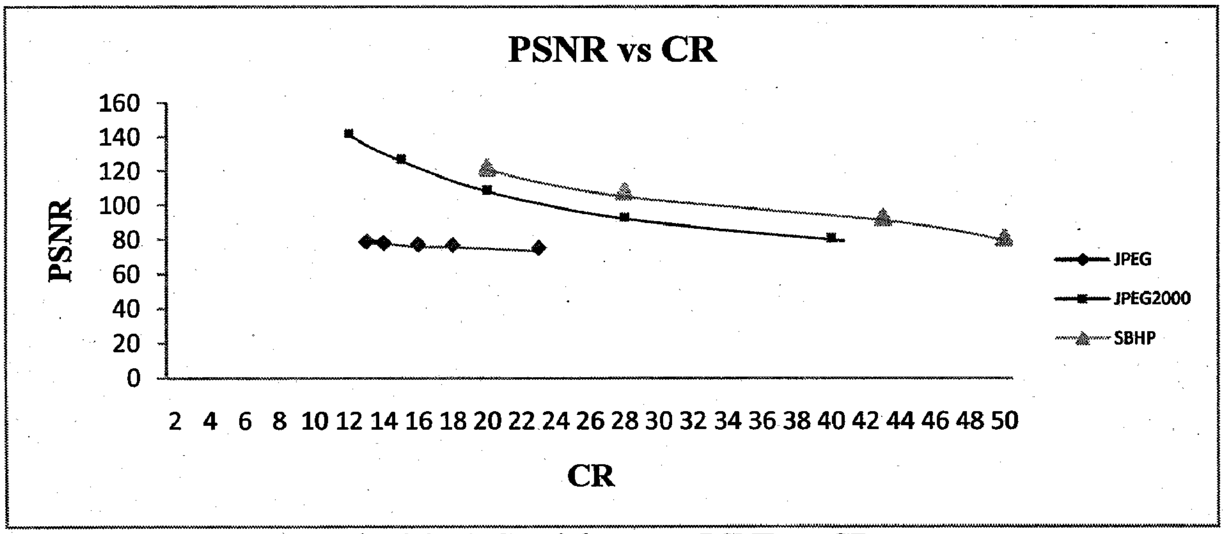


Fig 6.45 A Graph between PSNR vs CR

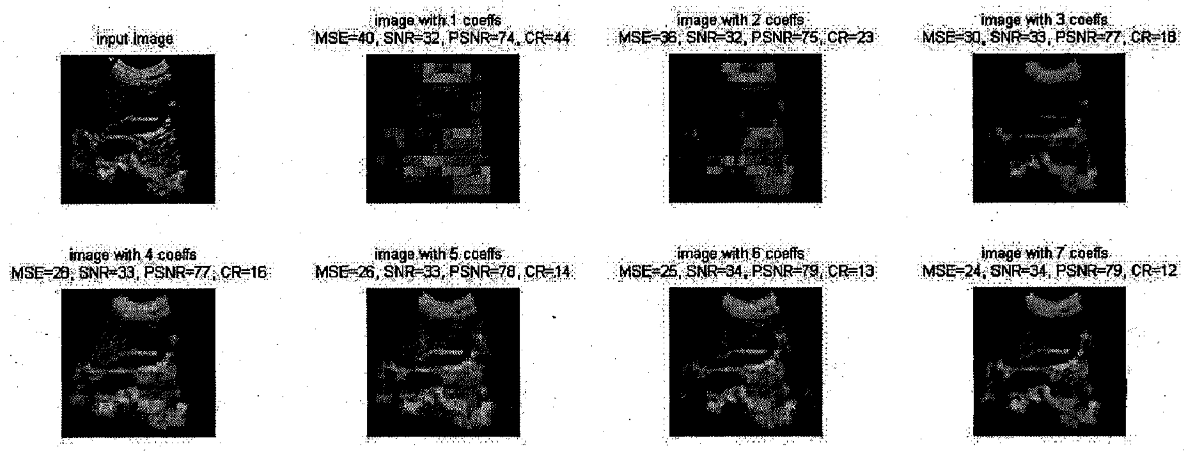


Fig 6.46 Output of JPEG

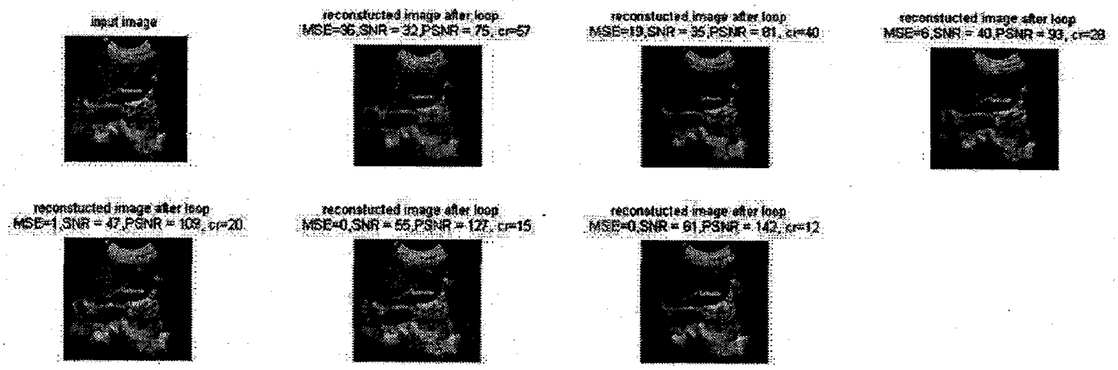


Fig 6.47 Output of JPEG 2000

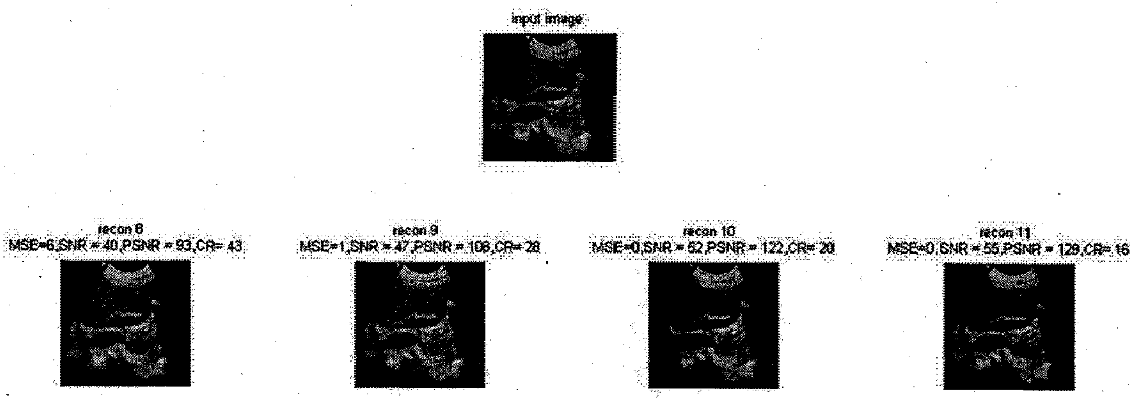


Fig 6.48 Output of SBHP

The three graphs are plotted from the table 6.22 & 6.23& 6.24 gives the following results.

From the graph 6.43, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.44, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.45, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 9:

Table 6.25.JPEG Output for Liver Image9:

coefficients	Mse	snr	Psnr	cr
2	41	31	74	24
3	35	32	75	18
4	32	33	76	16
5	29	33	77	14
6	25	34	79	12

Table 6.26.JPEG2000 Output for Liver Image9:

loop	Mse	snr	Psnr	cr
6	17	35	82	40
7	6	40	93	28
8	1	47	109	20
9	0	54	126	15
10	0	62	143	12

Table 6.27.SBHP Output for Liver Image9:

loop	Mse	snr	Psnr	cr
7	18	35	82	72
8	6	40	93	44
9	1	47	108	28
10	0	52	122	20
11	0	55	129	16

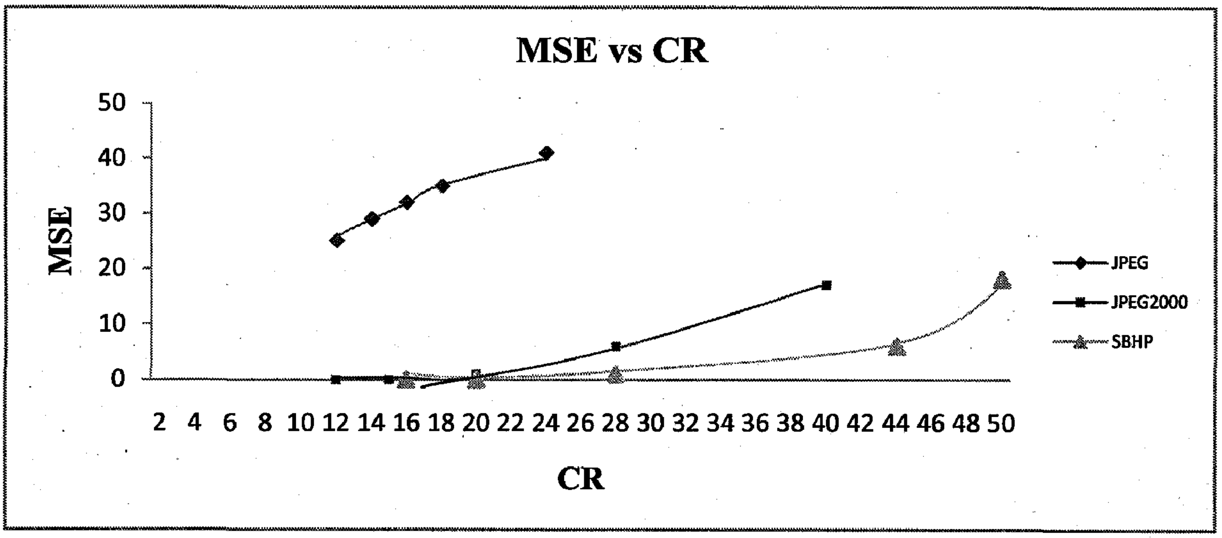


Fig 6.49 A Graph between MSE vs CR

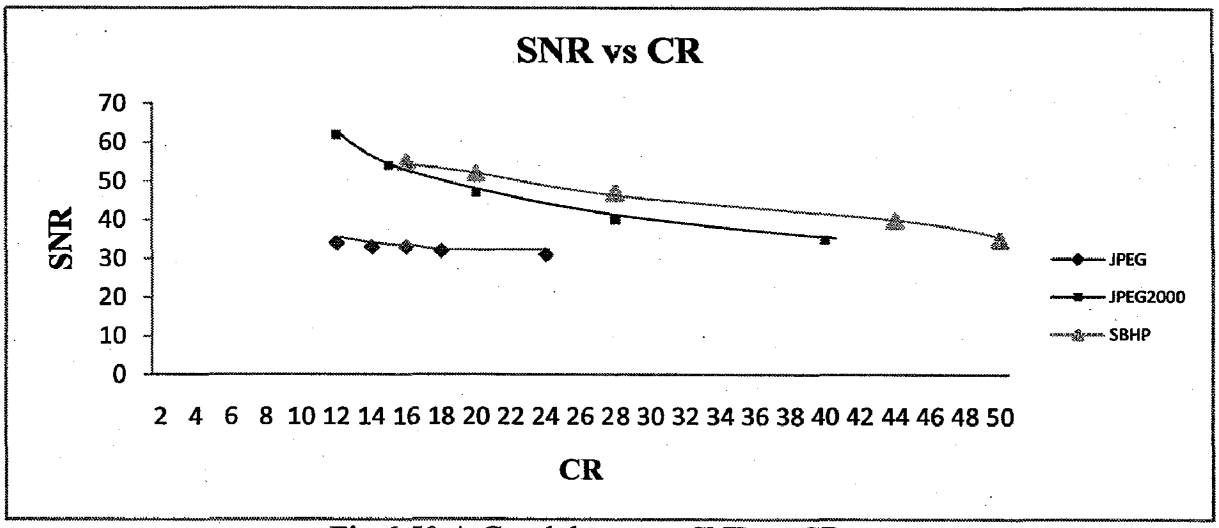


Fig 6.50 A Graph between SNR vs CR

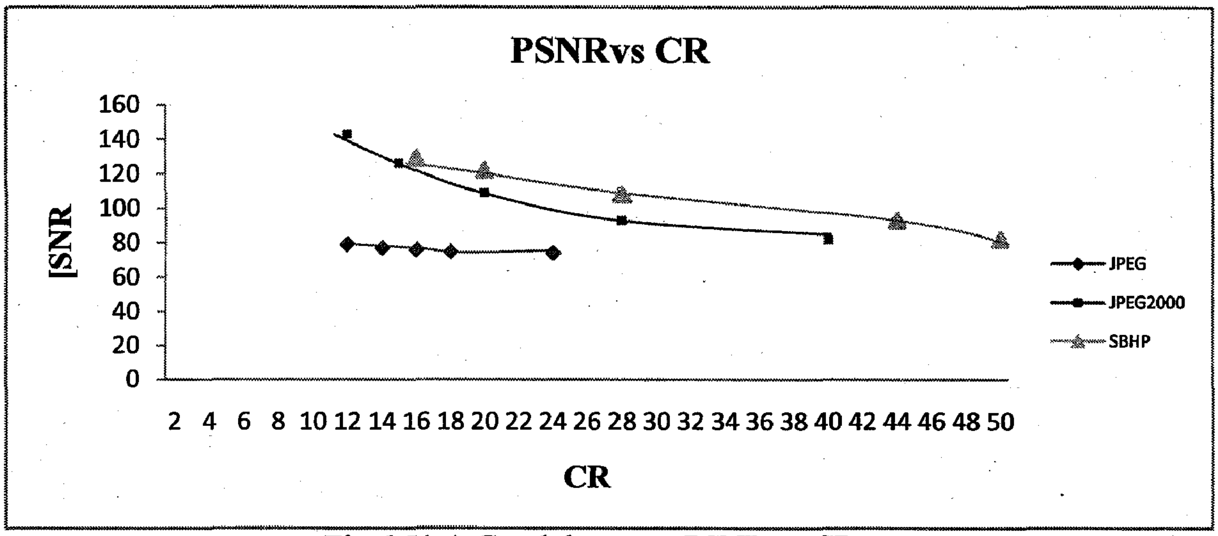


Fig 6.51 A Graph between PSNR vs CR

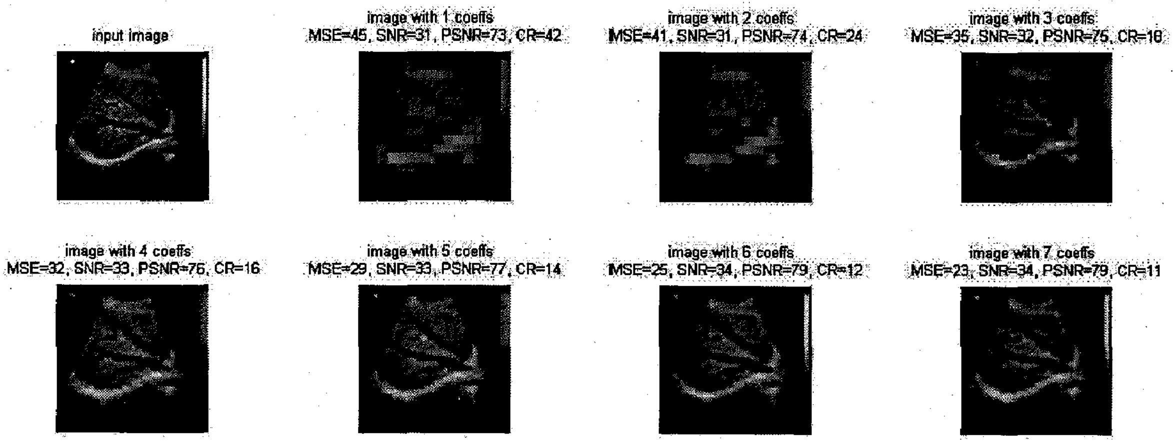


Fig 6.52 Output of JPEG

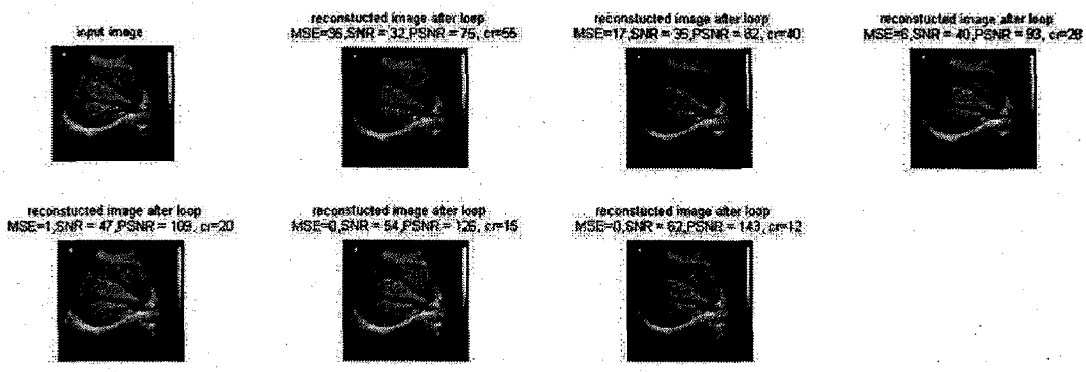


Fig 6.53 Output of JPEG 2000

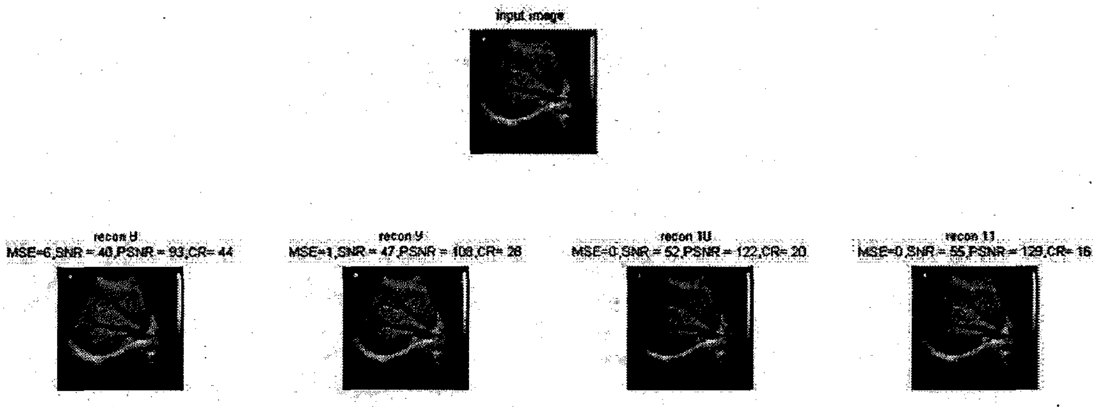


Fig 6.54 Output of SBHP

The three graphs are plotted from the table 6.25 & 6.26& 6.27 gives the following results.

From the graph 6.49, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.50, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.51, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Image 10:

Table 6.28.JPEG Output for Liver Image10:

coefficients	Mse	snr	Psnr	cr
1	68	29	69	**29**
2	61	30	70	**17**
3	58	30	70	**14**
4	56	30	71	**12**
5	52	30	71	**11**

Table 6.29.JPEG2000 Output for Liver Image10:

loop	Mse	snr	Psnr	cr
6	21	34	80	30
7	6	40	93	22
8	1	47	110	17
9	0	55	127	14
10	0	61	143	11

Table 6.30.SBHP Output for Liver Image10:

loop	Mse	snr	Psnr	cr
7	22	34	80	42
8	7	39	92	28
9	1	46	108	21
10	0	53	122	16
11	0	56	129	13

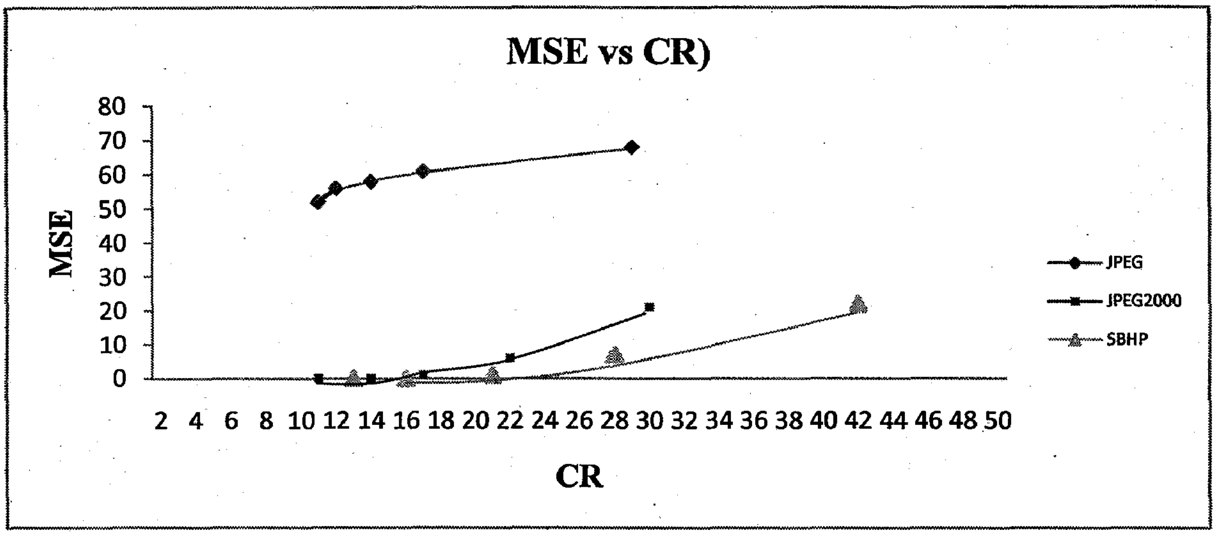


Fig 6.55 A Graph between MSE vs CR

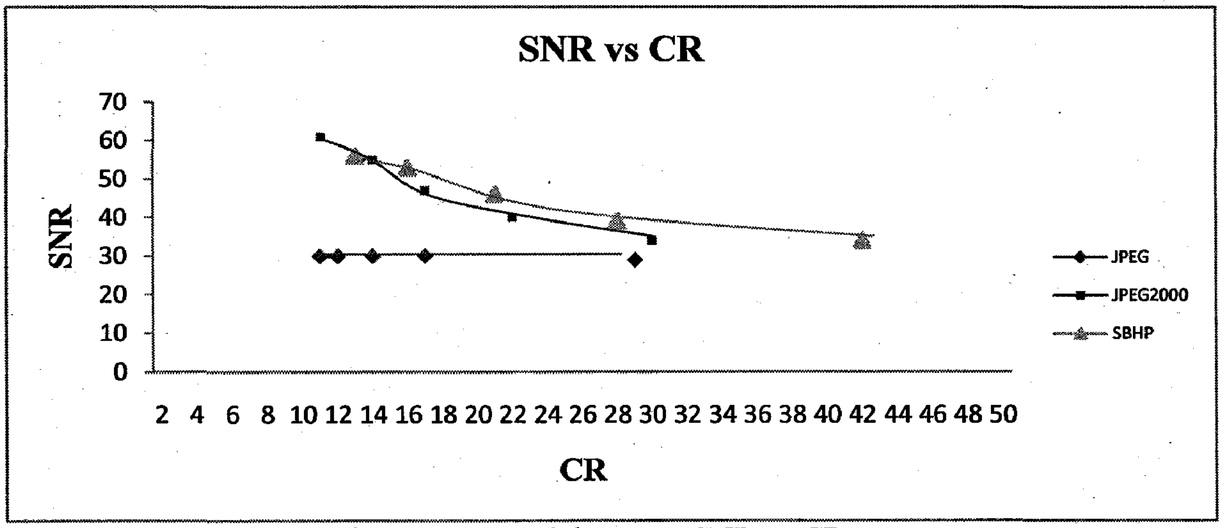


Fig 6.56 A Graph between SNR vs CR

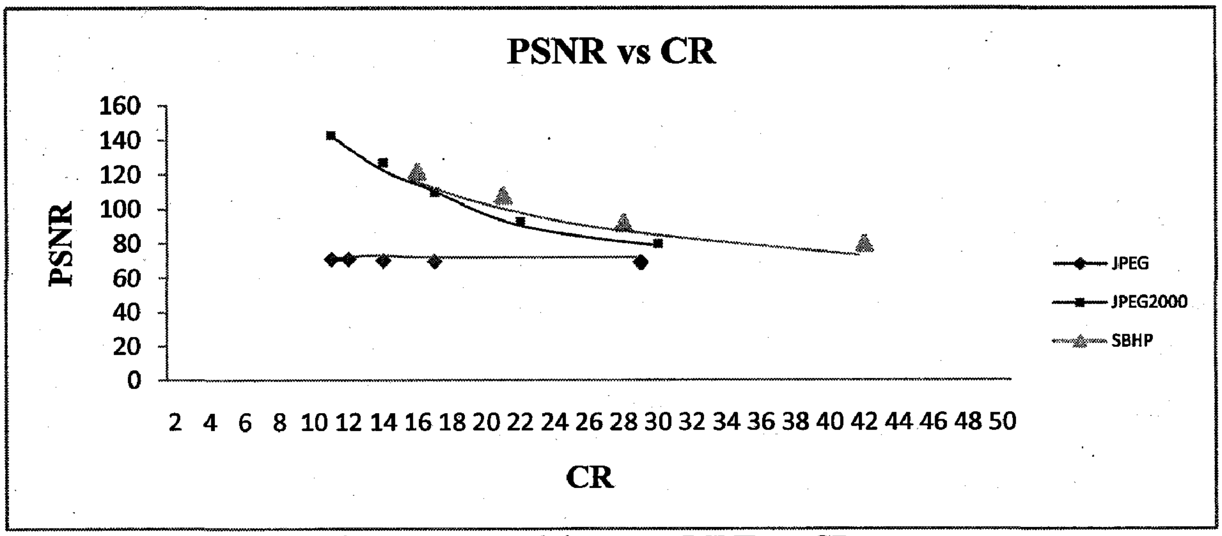


Fig 6.57 A Graph between PSNR vs CR

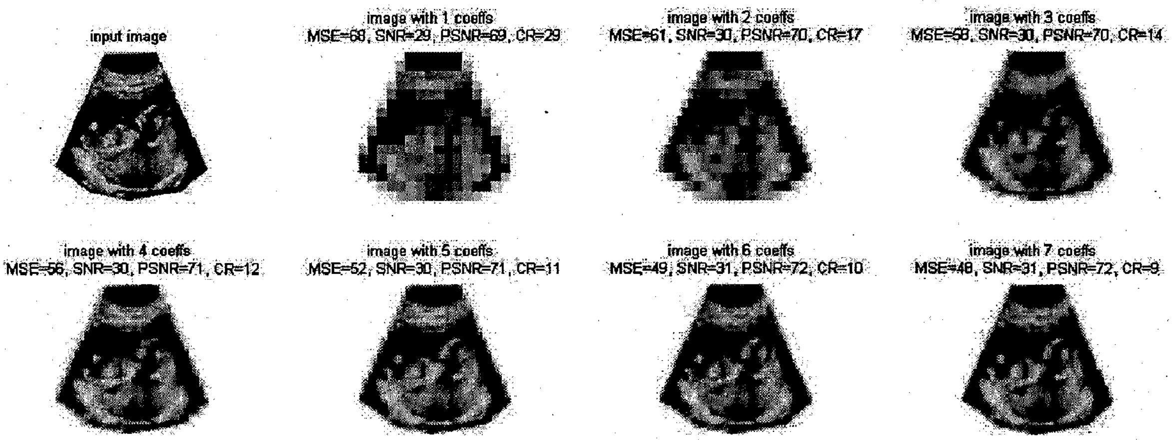


Fig 6.58 Output of JPEG

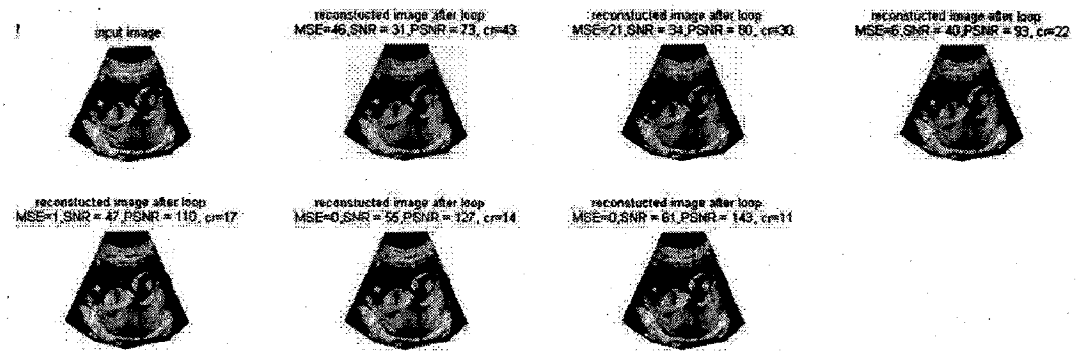


Fig 6.59 Output of JPEG 2000

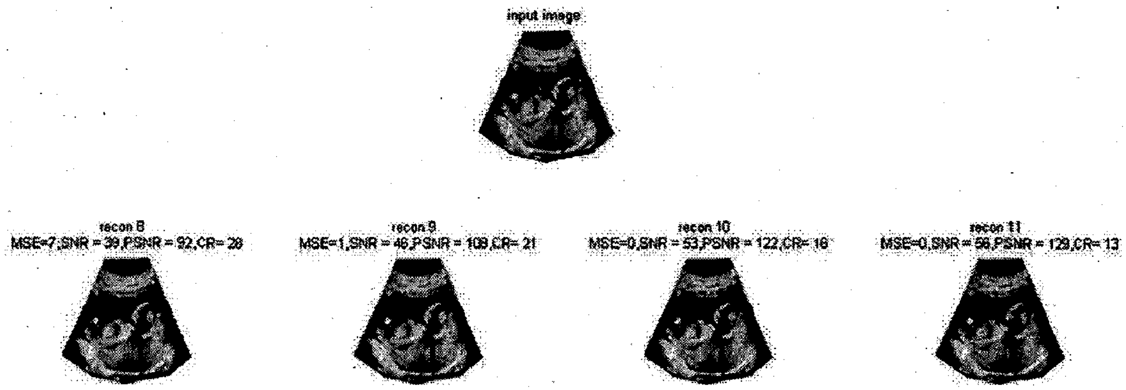


Fig 6.60 Output of SBHP

The three graphs are plotted from the table 6.28 & 6.29& 6.30 gives the following results.

From the graph 6.55, it is observed that, MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.56, it is observed that, SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

From the graph 6.57, it is observed that, PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

6.2 Results on Kidney Ultrasound images:

Image1:

Table6.31 JPEG Output for kidney image1:

coefficients	Mse	snr	Psnr	cr
5	29	33	77	14
6	28	33	78	12
7	28	33	78	12
8	26	33	78	11
9	25	34	79	10

Table6.32JPEG2000 Output for kidney image1:

loop	Mse	snr	Psnr	cr
6	19	35	81	39
7	6	40	93	27
8	1	47	109	19
9	0	54	126	14
10	0	61	143	12

Table6.33SBHP Output for kidney image1:

loop	Mse	snr	Psnr	cr
7	20	35	81	65
8	7	39	92	40
9	1	46	108	26
10	0	52	121	19
11	0	55	128	15

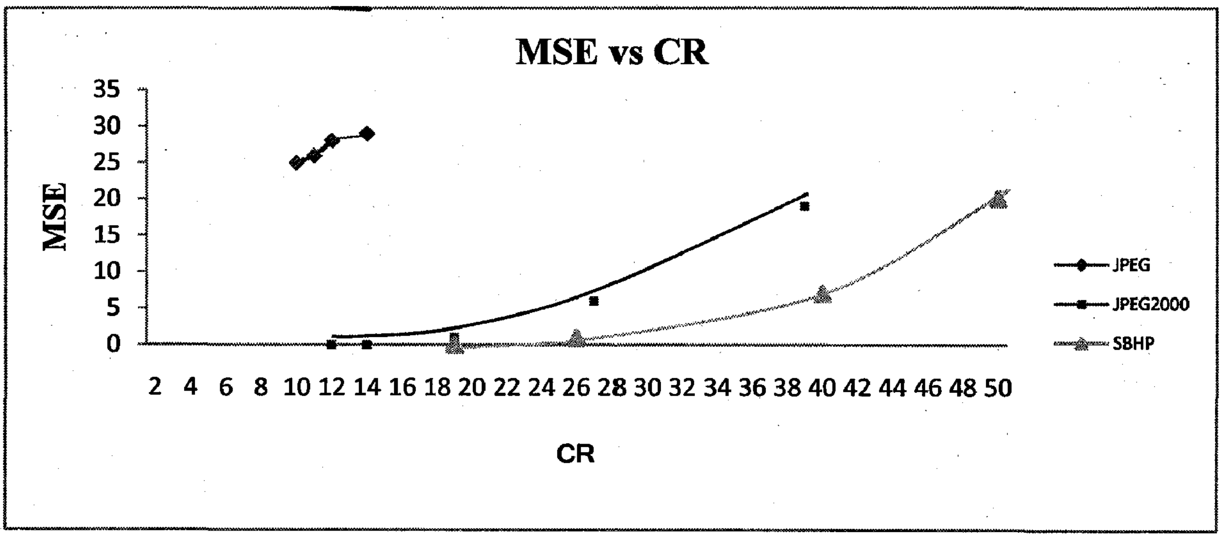


Fig.6.61 A graph between MSE vs. CR

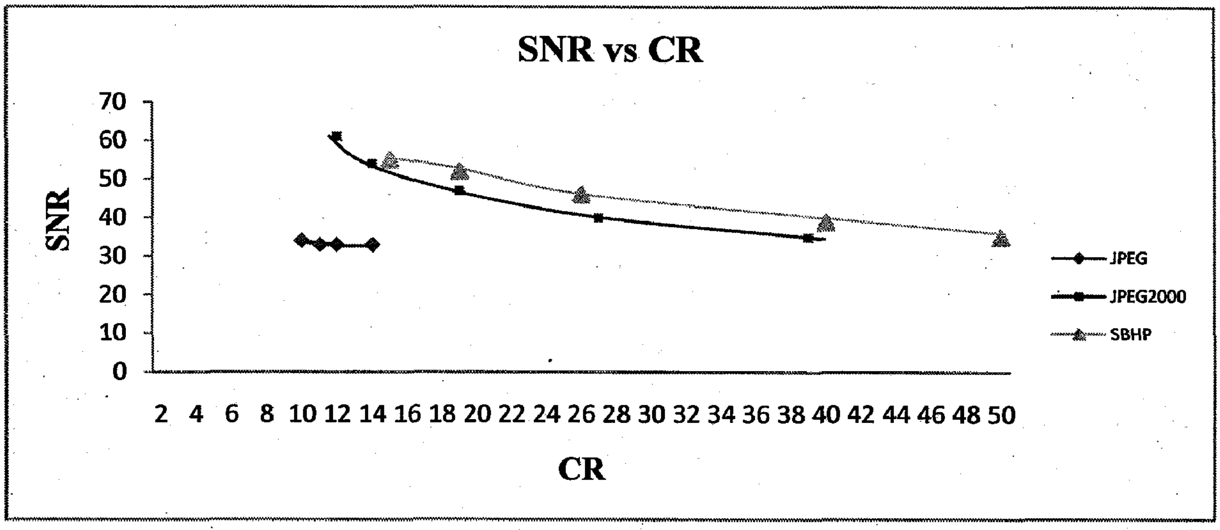


Fig 6.62 A graph between SNRvs CR

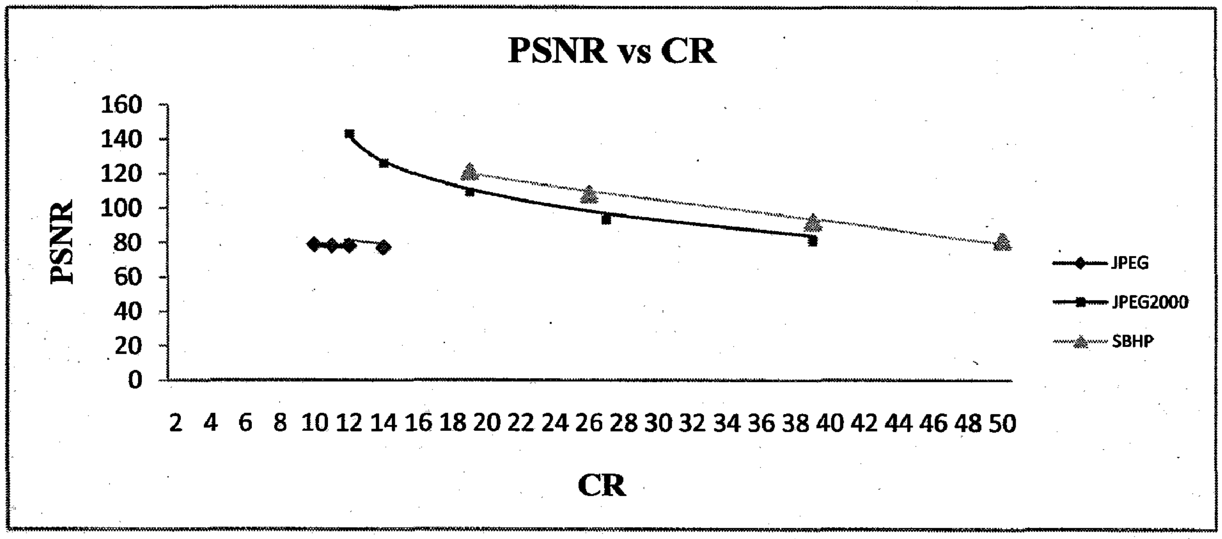


Fig.6.63 A graph between PSNR vs CR

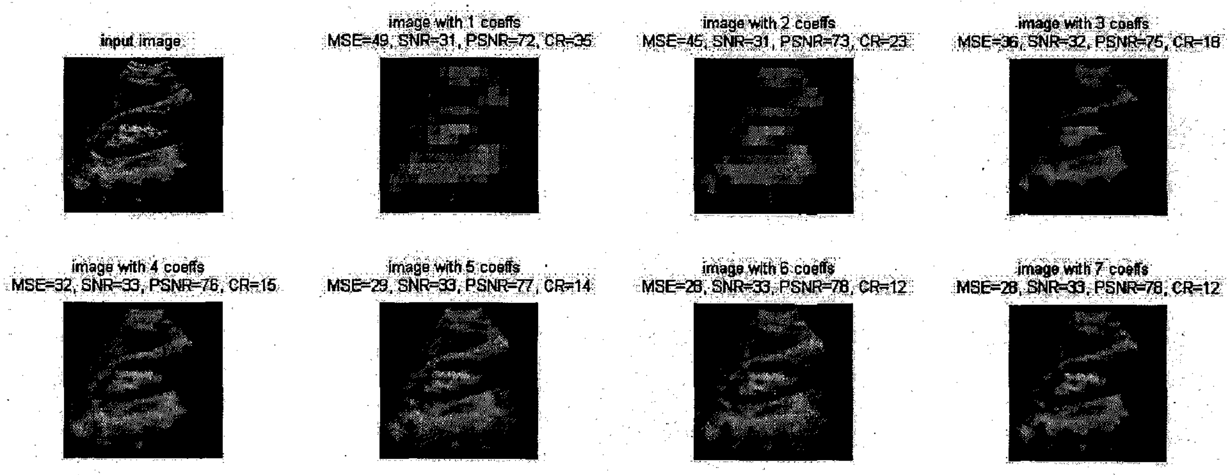


Fig.6.64 Output of JPEG

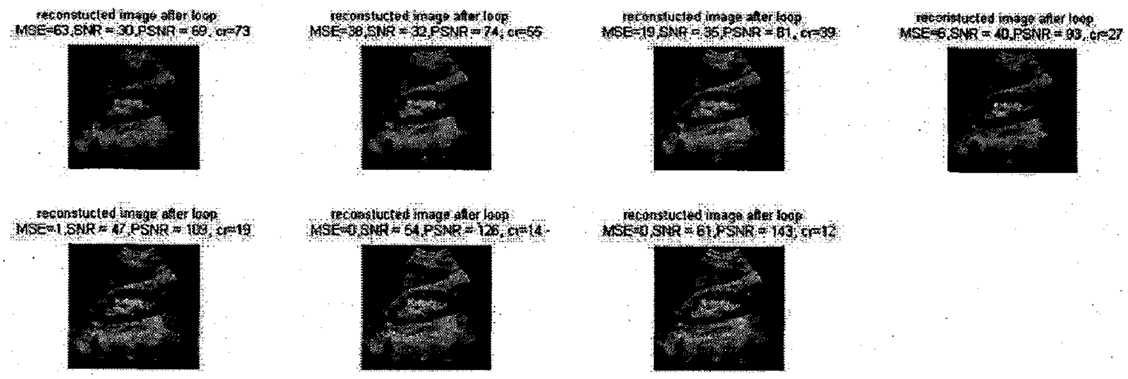


Fig 6.65 Output of JPEG2000

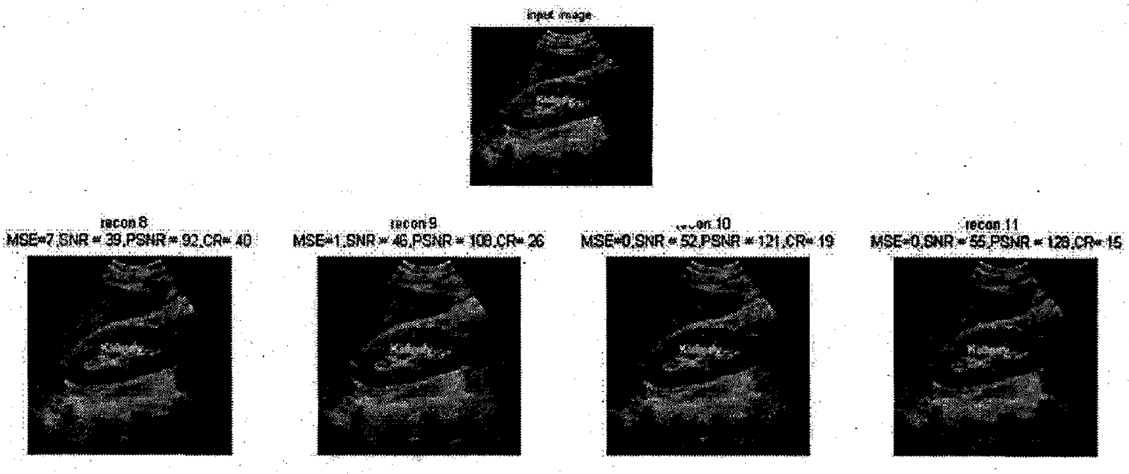


Fig.6.66 Output of SBHP

Image 2:

Table6.34.JPEG Output for kidney image2:

coefficients	Mse	snr	Psnr	cr
5	21	34	80	15
6	21	34	81	13
7	20	35	81	13
8	19	35	81	12
9	18	35	82	11

Table6.35.JPEG2000 Output for kidney image2:

loop	Mse	snr	Psnr	cr
6	17	35	83	44
7	5	40	94	30
8	1	47	110	21
9	0	55	127	16
10	0	61	142	13

Table6.36.SBHP Output for kidney image2:

loop	Mse	snr	Psnr	cr
7	17	35	83	87
8	6	40	93	50
9	1	47	109	30
10	0	53	122	21
11	0	55	129	17

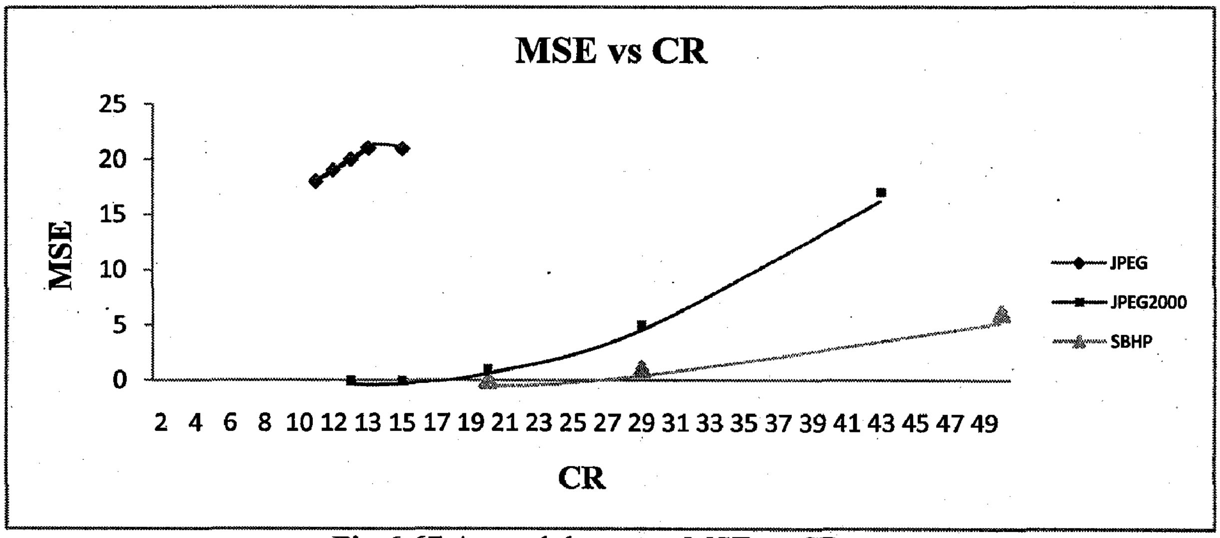


Fig.6.67 A graph between MSE vs CR

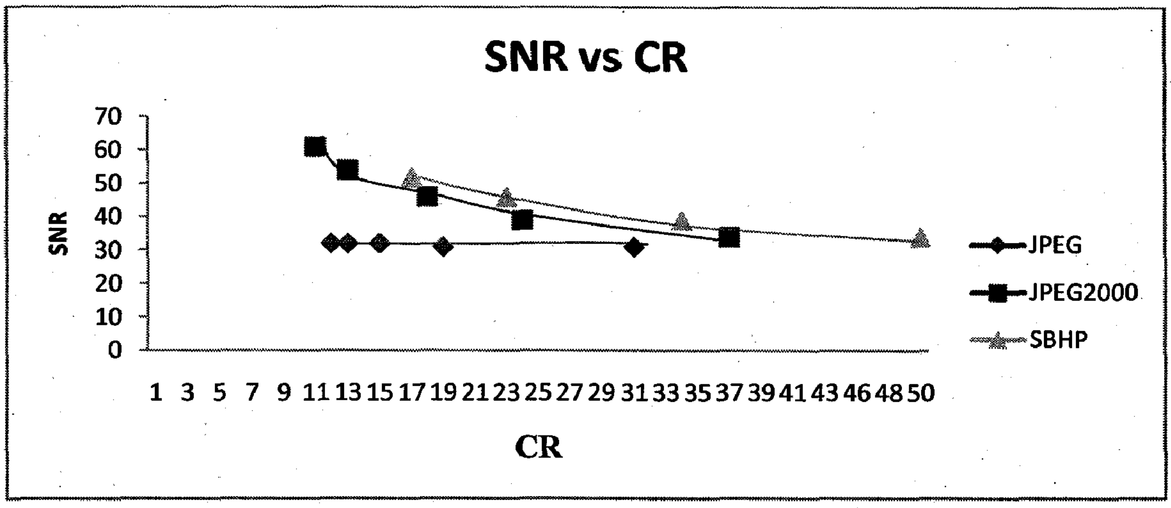
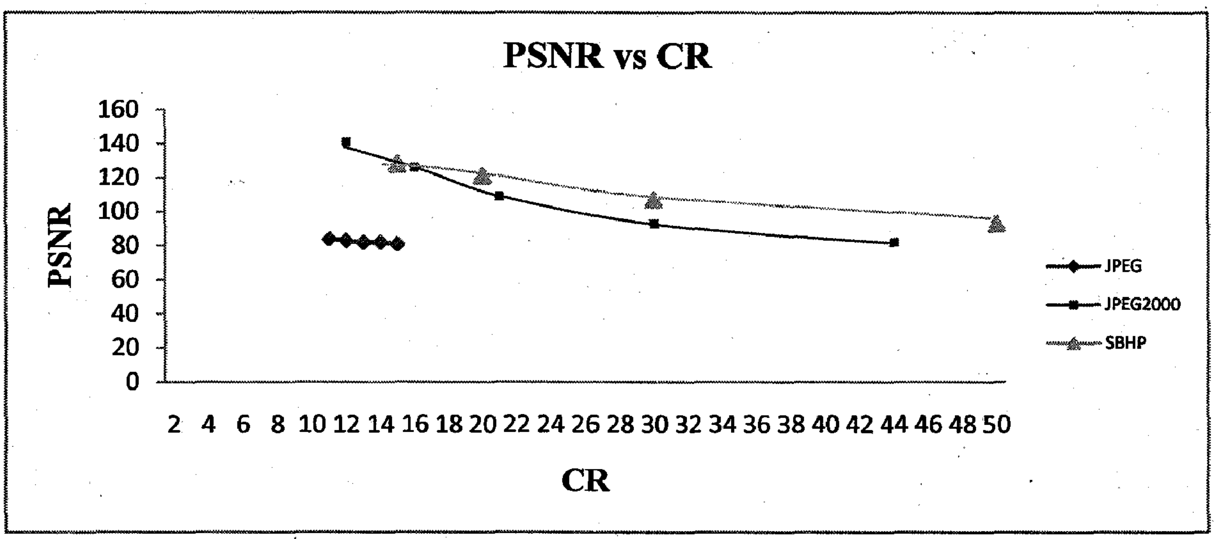


Fig.6.68 A graph between SNR vs CR



Fign6.69 A graph between PSNRvs CR

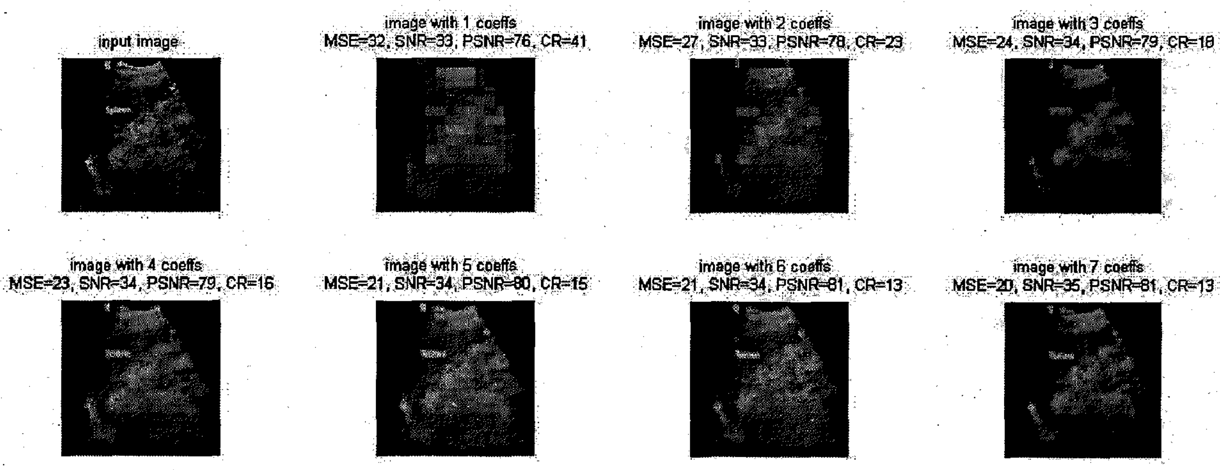


Fig.6.70 Output of JPEG

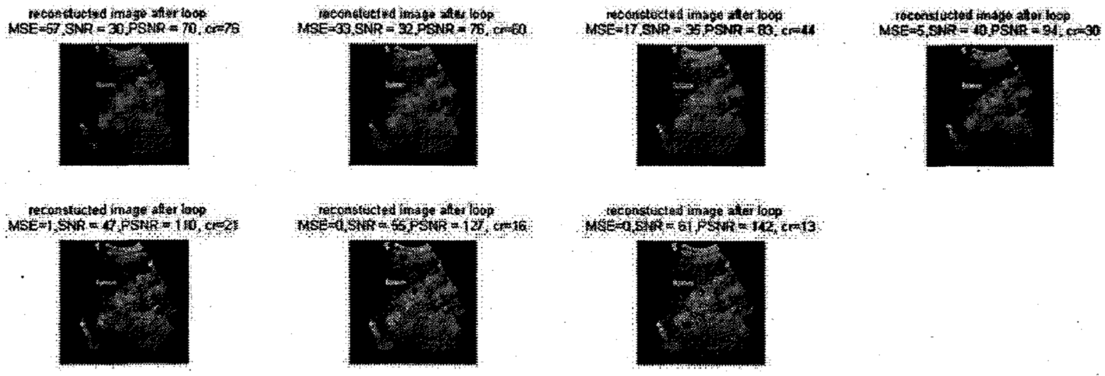


Fig.6.71 Output of JPEG2000

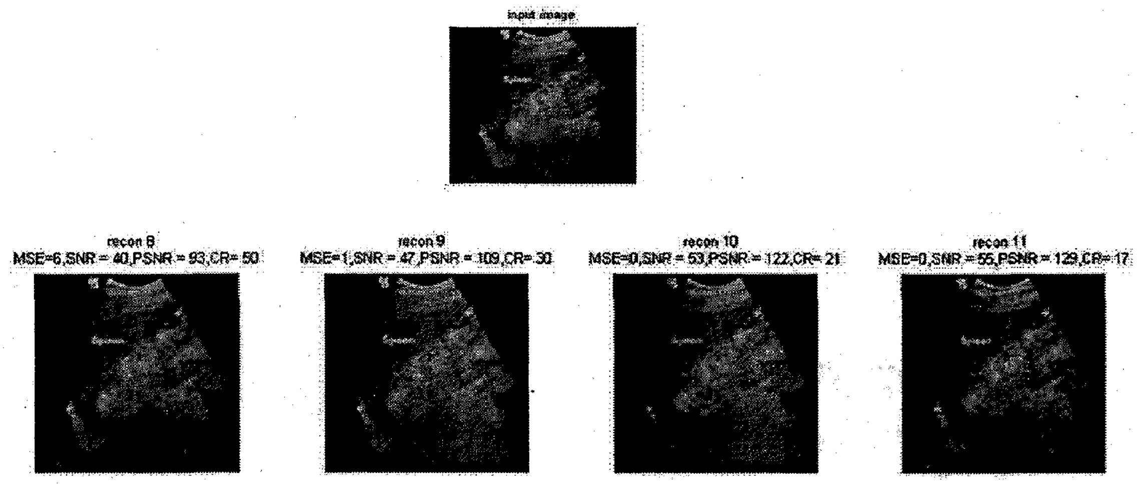


Fig.6.72 Output of SBHP

Image3:

Table6.37.JPEG Output for kidney image3:

coefficients	Mse	snr	Psnr	cr
5	19	35	81	15
6	18	35	82	14
7	18	35	82	13
8	16	35	83	12
9	14	36	84	11

Table6.38.JPEG2000 Output for kidney image3:

loop	Mse	snr	Psnr	cr
6	18	35	82	44
7	6	40	93	30
8	1	47	109	21
9	0	54	126	16
10	0	61	141	12

Table6.39.SBHP Output for kidney image3:

loop	Mse	Snr	Psnr	cr
7	18	35	82	87
8	6	40	93	50
9	1	46	107	30
10	0	52	121	20
11	0	55	128	15

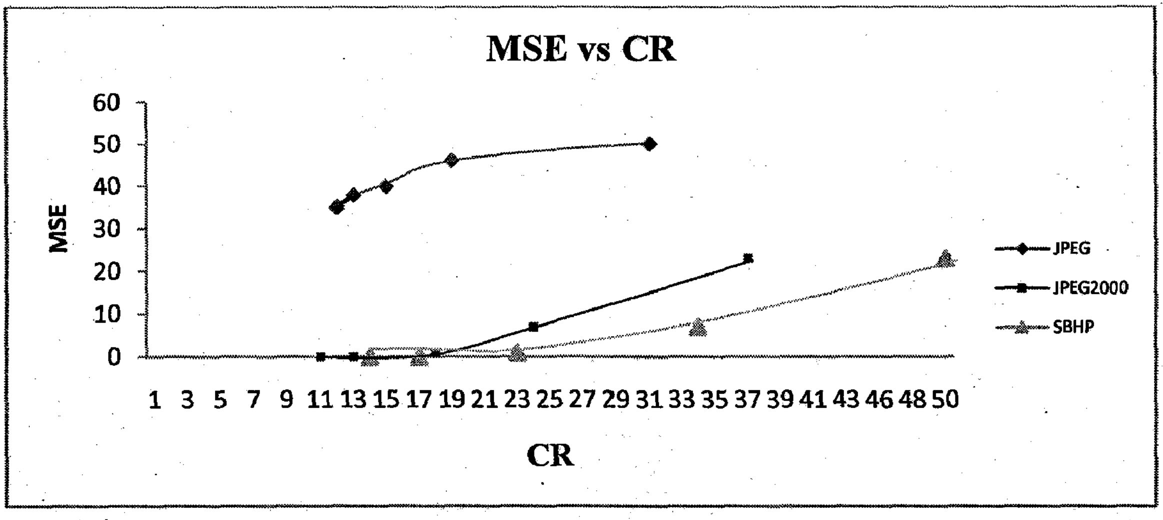


Fig.6.73 A graph between MSE vs CR

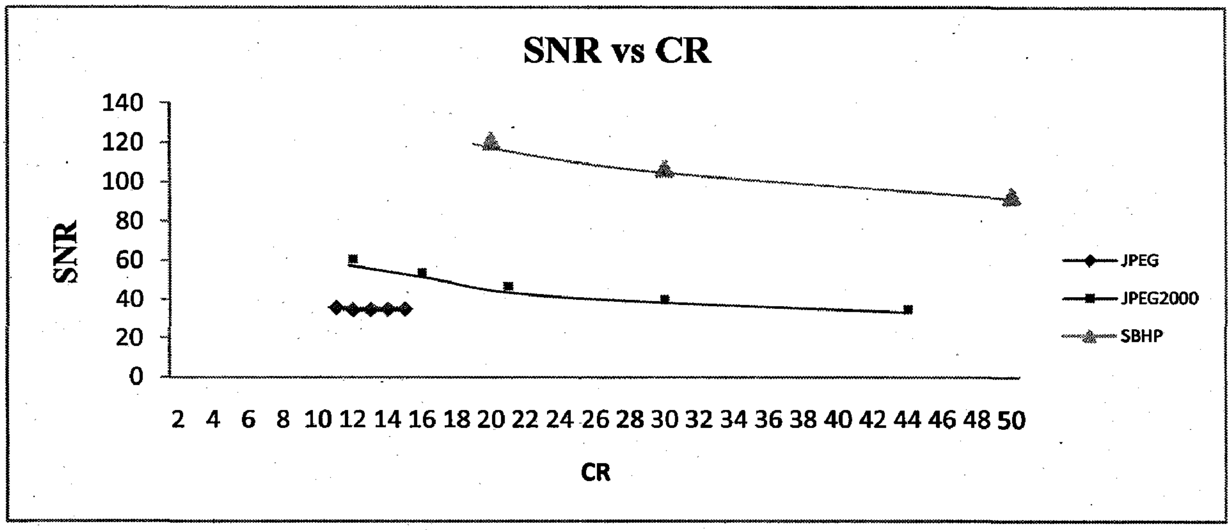


Fig.6.74 A graph between SNR vs CR

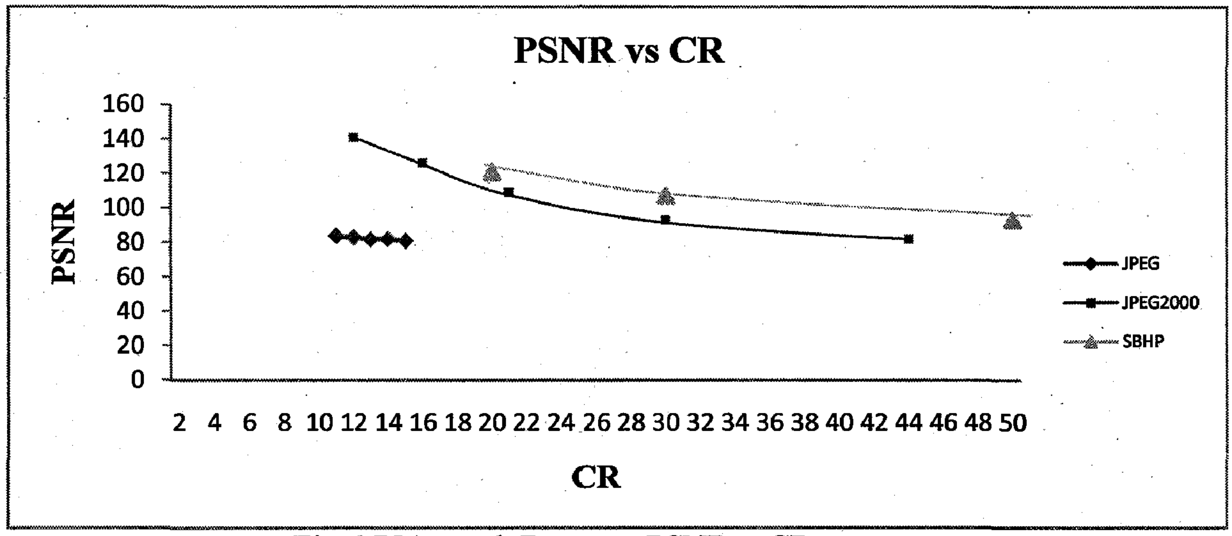


Fig.6.75A graph Between PSNRvs CR

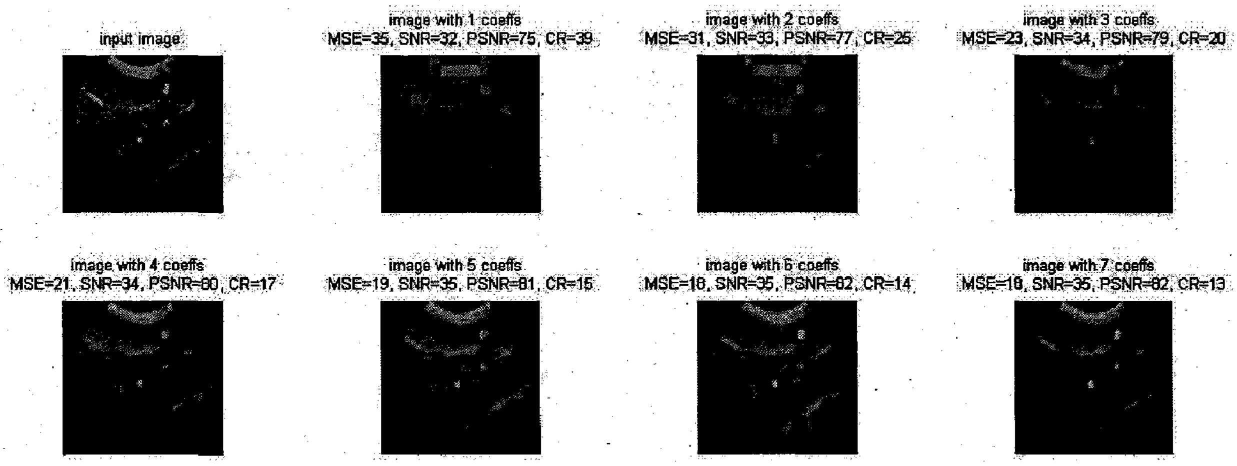


Fig.6.76 Output of JPEG

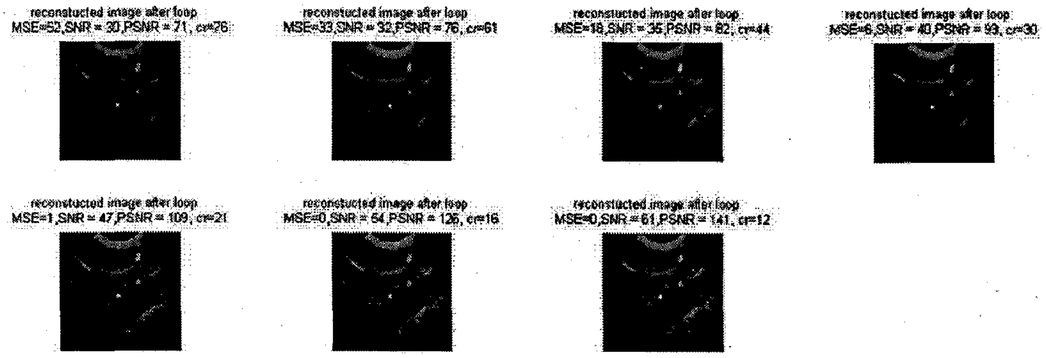


Fig.6.77 Output of JPEG2000

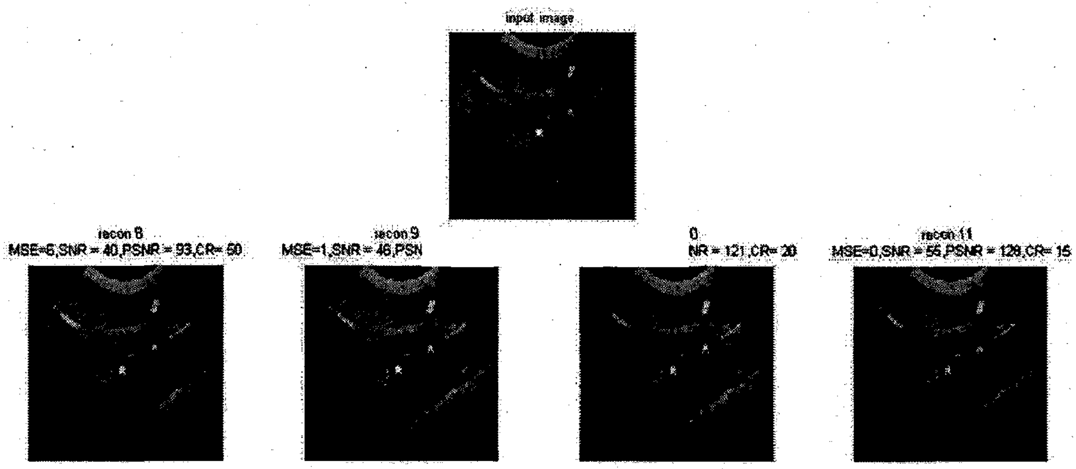


Fig.6.78 Output of SBHP

Image4:

Table6.403.JPEG Output for kidney image4:

coefficients	Mse	snr	Psnr	cr
4	39	32	74	16
5	36	32	75	14
6	35	32	75	13
7	35	32	75	12
8	33	32	76	11

Table6.41.JPEG2000 Output for kidney image4:

loop	Mse	snr	Psnr	cr
6	24	34	79	38
7	7	39	91	25
8	1	46	108	18
9	0	54	125	13
10	0	61	141	11

Table6.42.SBHP Output for kidney image4:

loop	Mse	snr	Psnr	cr
7	24	34	79	61
8	8	39	90	35
9	2	46	106	22
10	0	52	121	17
11	0	55	127	13

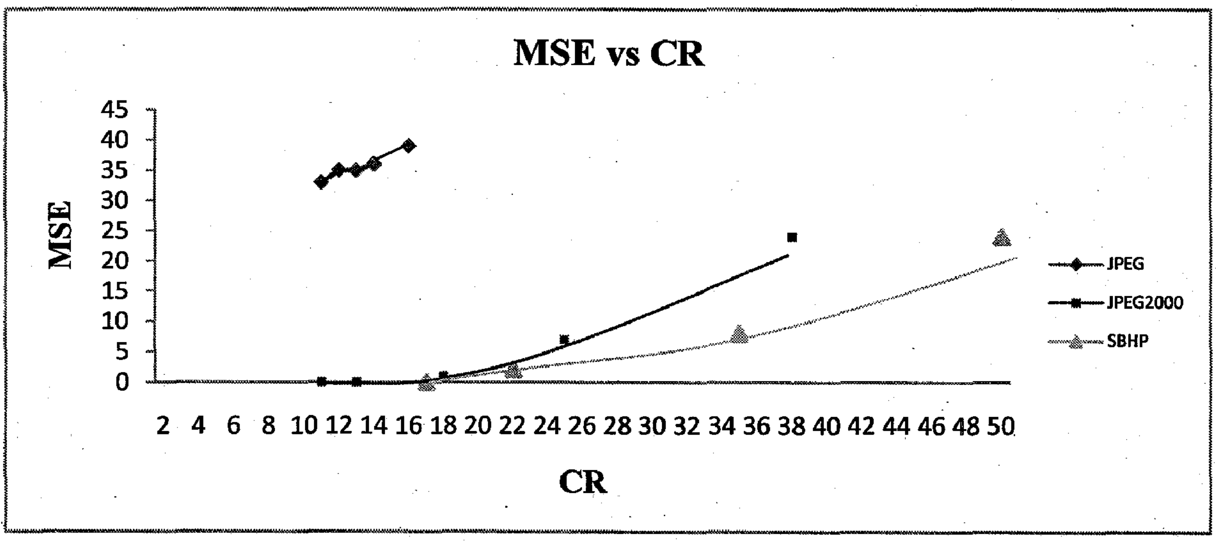


Fig.6.79 A graph between MSE vs CR

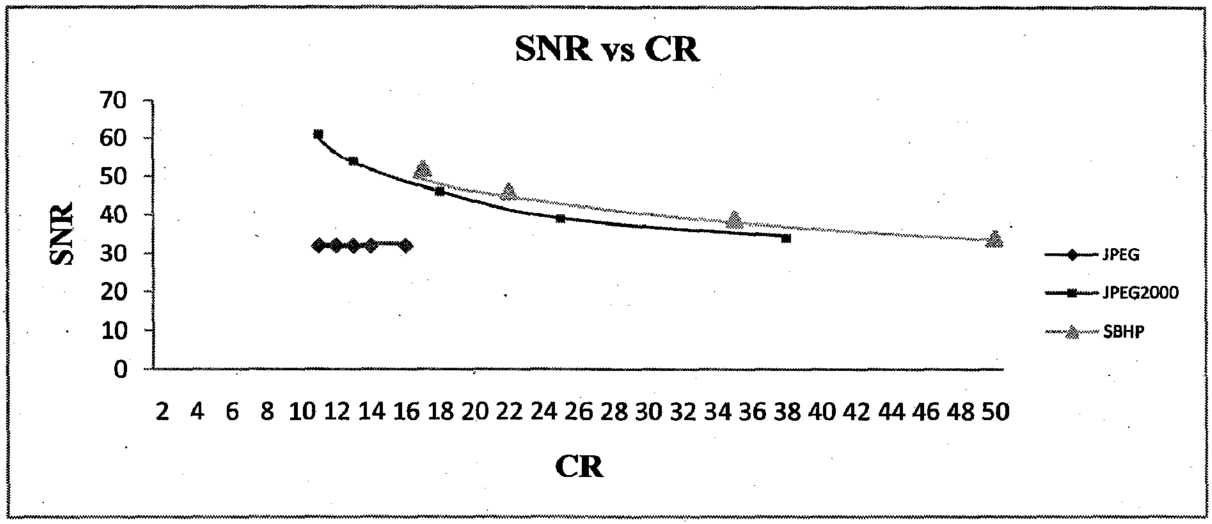


Fig.6.80 A graph between SNR vs CR

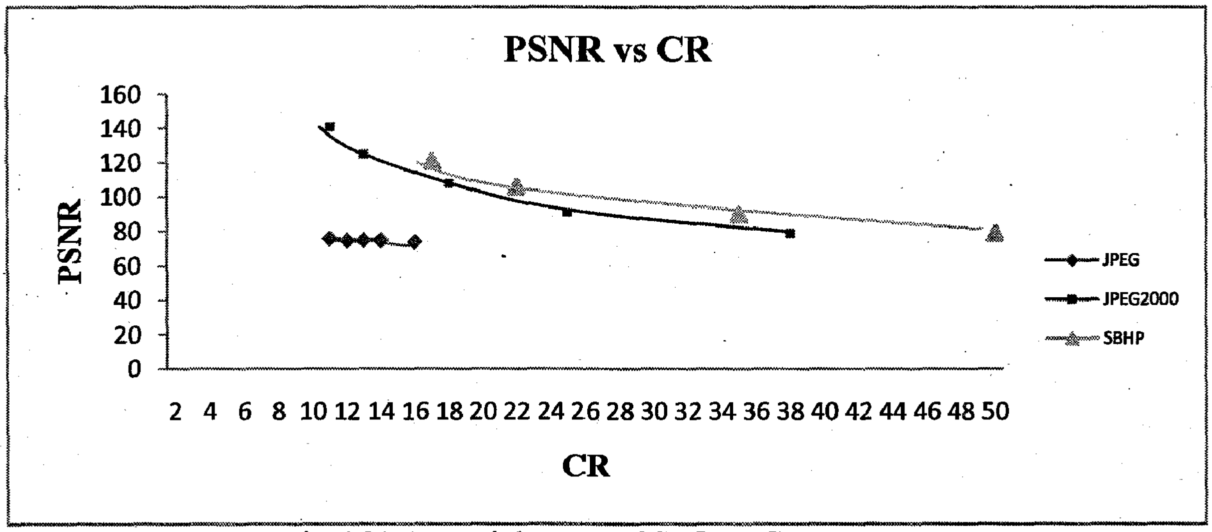


Fig.6.81 A graph between PSNR vs CR

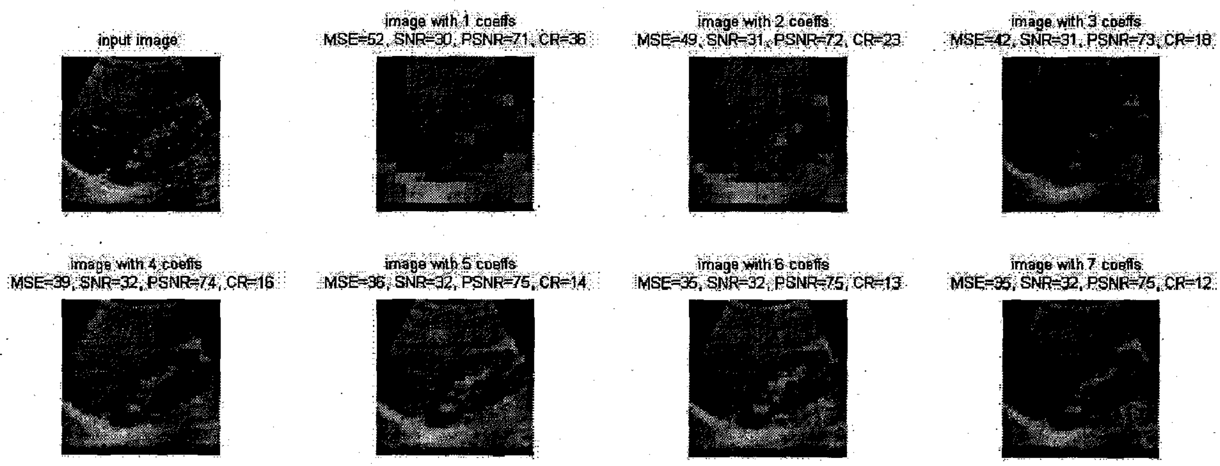


Fig.6.82 Output of JPEG

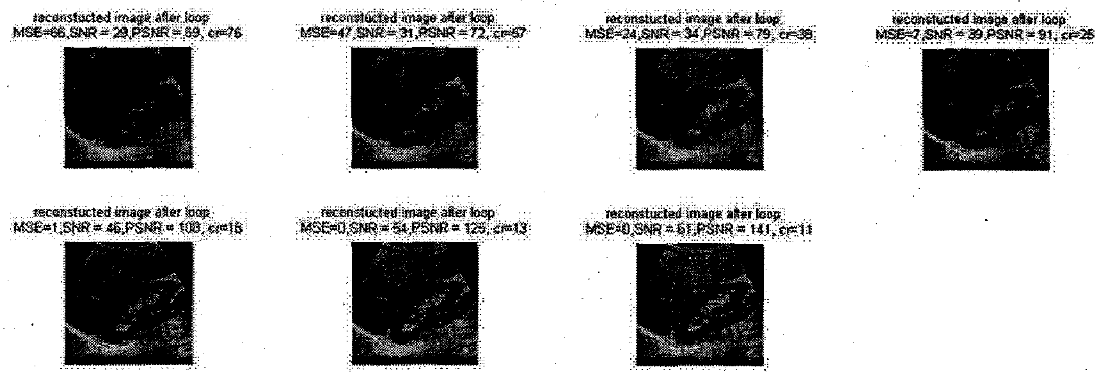


Fig.6.83 Output of JPEG2000

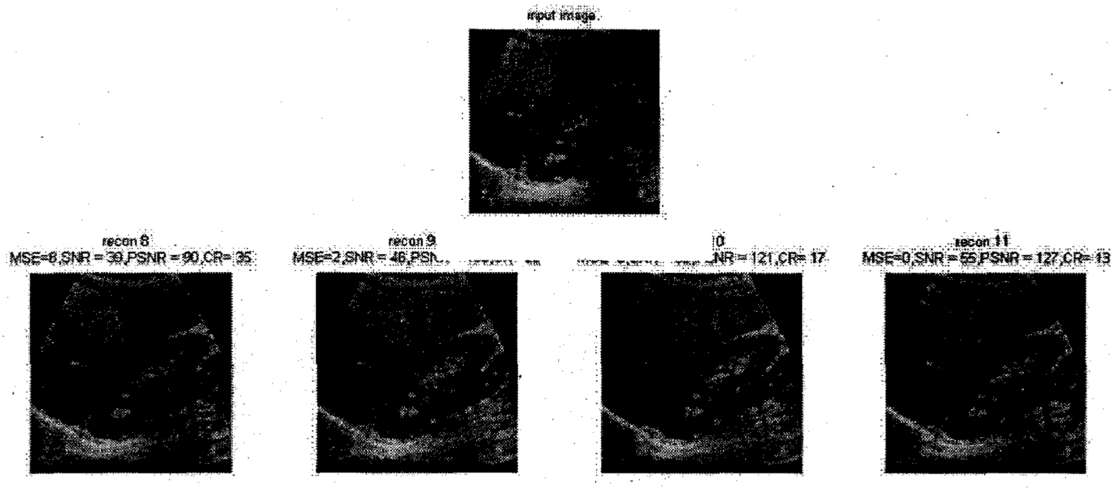


Fig.6.84 Output of SBHP

Image5:

Table6.43.JPEG Output for kidney image5:

coefficients	Mse	snr	Psnr	cr
5	32	33	76	14
6	31	33	76	13
7	30	33	77	12
8	29	33	77	11
9	28	33	78	10

Table6.44.JPEG2000 Output for kidney image5:

loop	Mse	snr	Psnr	cr
6	23	34	79	39
7	7	39	91	26
8	1	46	107	18
9	0	54	125	13
10	0	61	141	11

Table6.45.SBHP Output for kidney image5:

loop	Mse	snr	Psnr	cr
7	23	34	79	67
8	8	39	91	38
9	2	46	106	23
10	0	52	121	17
11	0	55	127	14

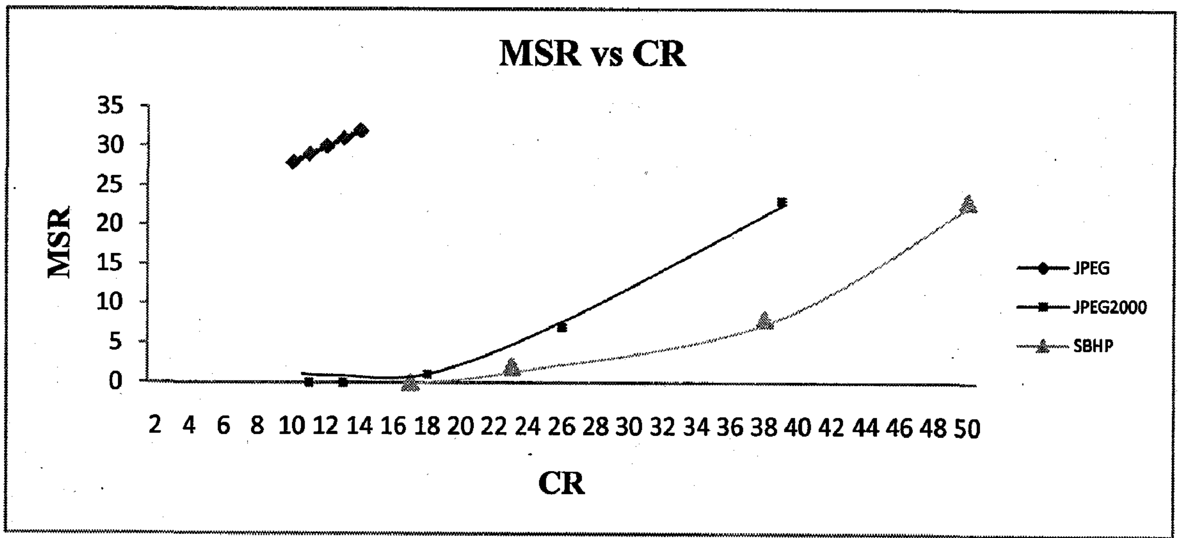


Fig.6.85 A graph between MSR vs CR

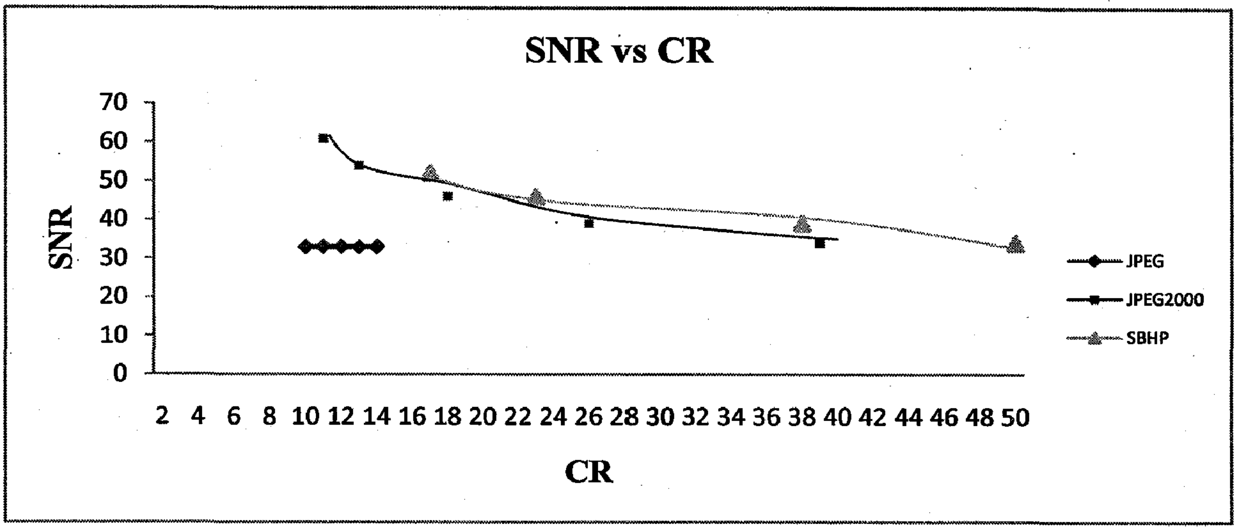


Fig.6.86 A graph between SNR vs CR

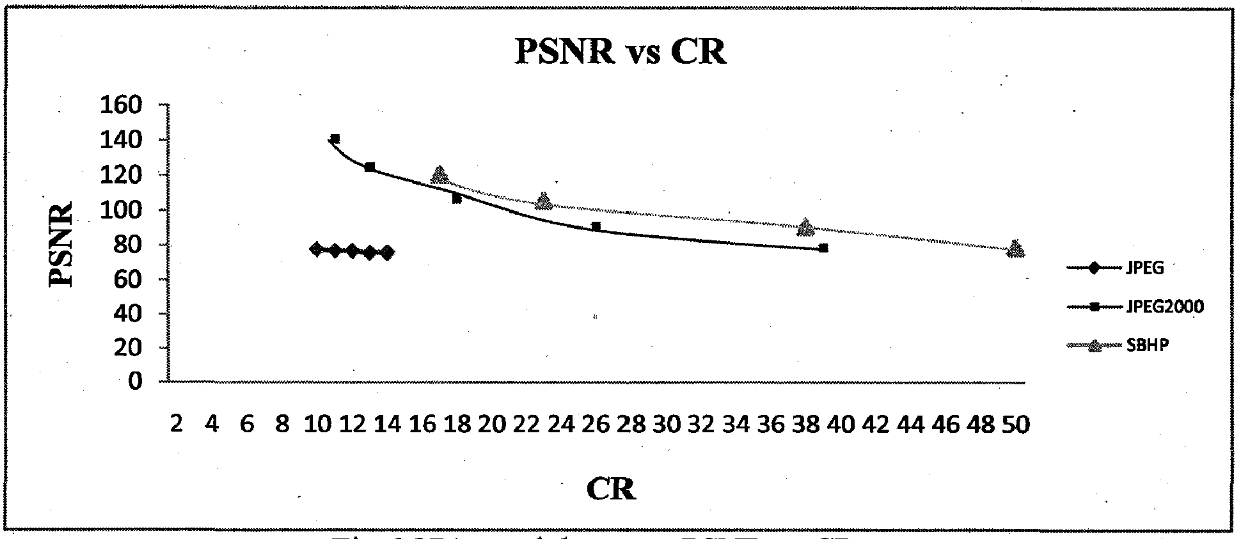


Fig.6.87A graph between PSNR vs CR

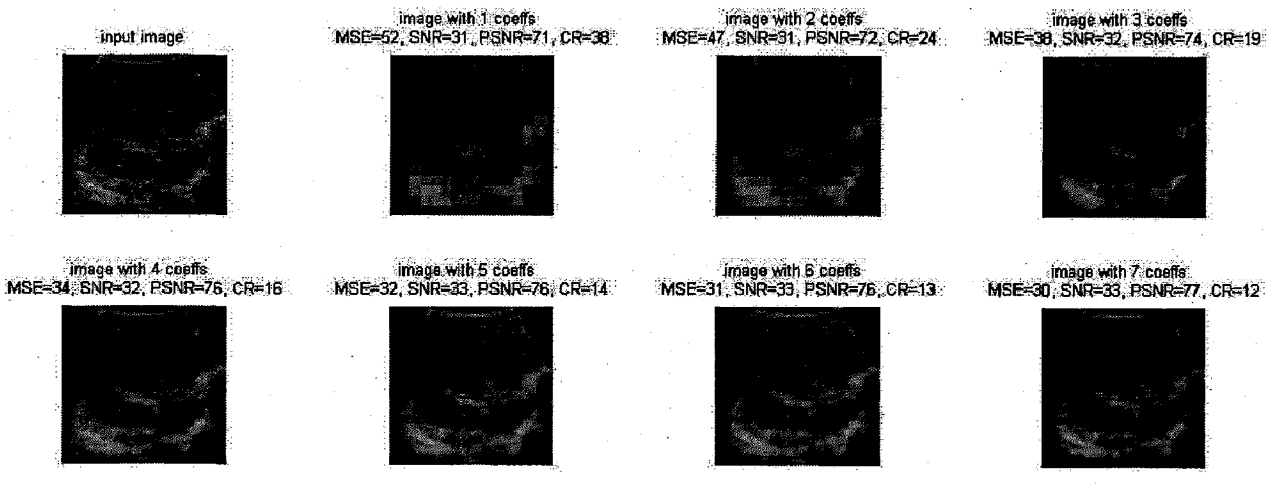


Fig.6.88 Output of JPEG

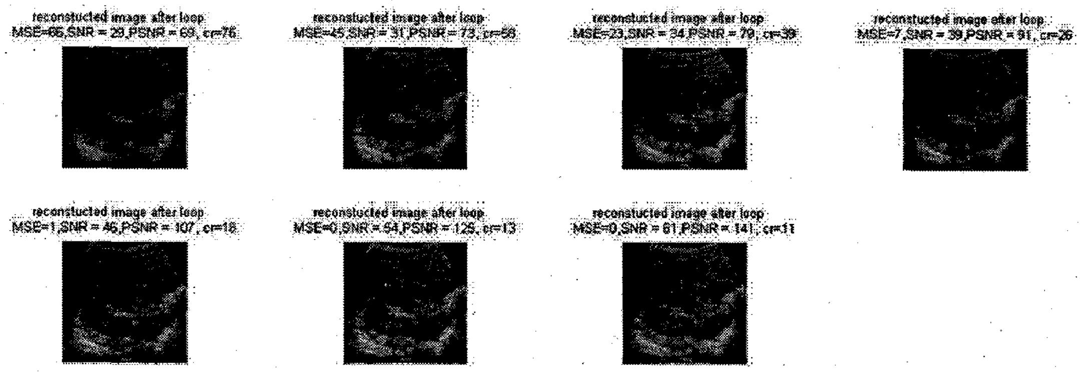


Fig.6.89 Output of JPEG2000

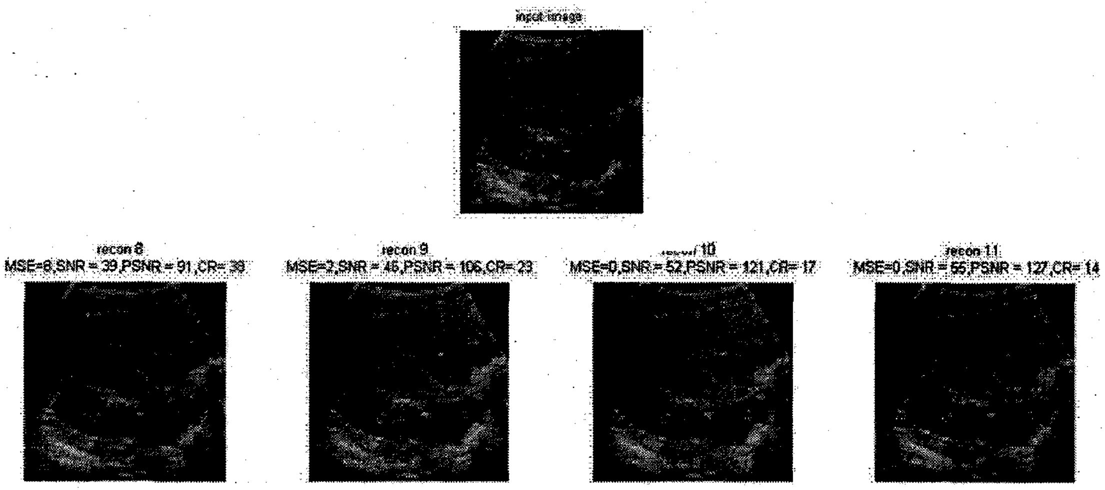


Fig.6.90 Output of SBHP

Image6:

Table6.46.JPEG Output for kidney image6:

coefficients	Mse	snr	Psnr	cr
3	43	31	73	17
4	41	32	74	14
5	38	32	74	13
6	37	32	75	12
7	36	32	75	11

Table6.47.JPEG2000 Output for kidney image6:

loop	Mse	snr	Psnr	cr
6	24	34	79	36
7	7	39	91	24
8	1	46	108	17
9	0	54	125	13
10	0	61	141	11

Table6.48.SBHP Output for kidney image6:

loop	Mse	snr	Psnr	cr
7	25	34	79	54
8	8	39	90	33
9	2	46	106	22
10	0	52	120	16
11	0	55	127	13

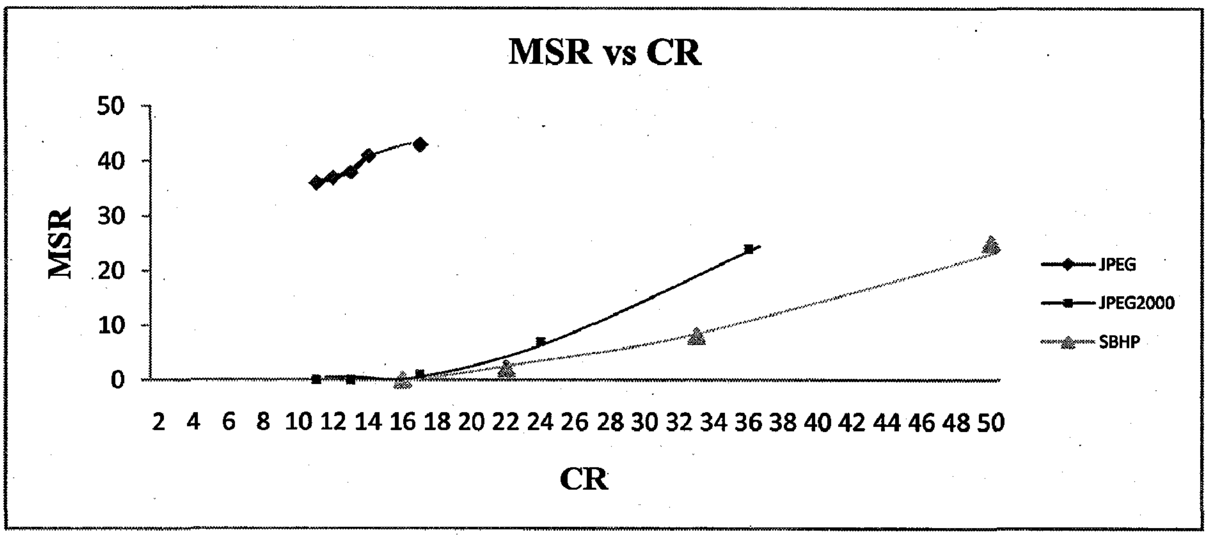


Fig.6.91 A graph between MSR vs CR

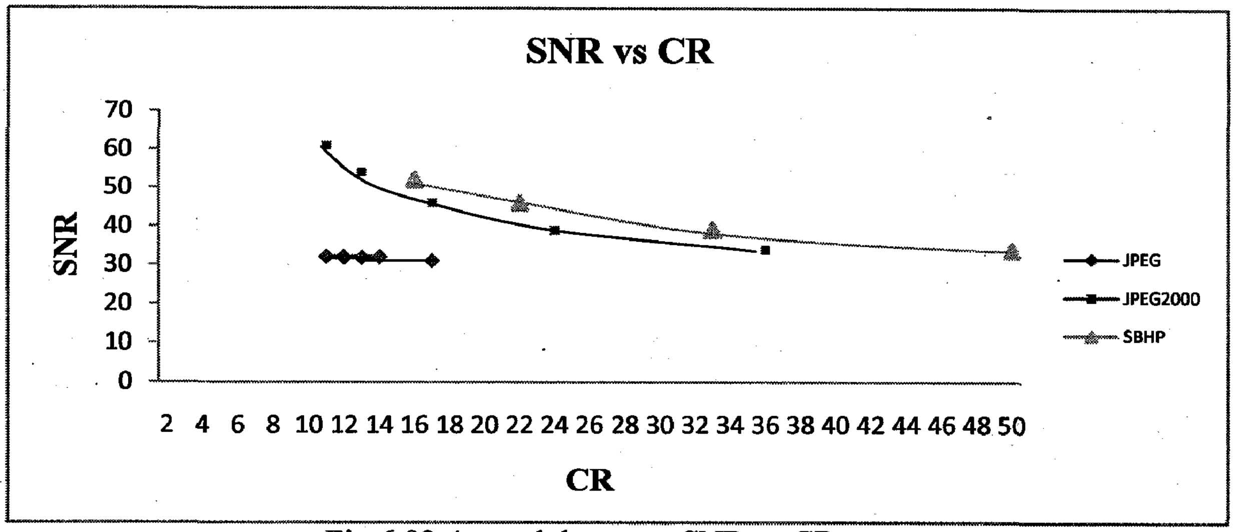


Fig.6.92 A graph between SNR vs CR

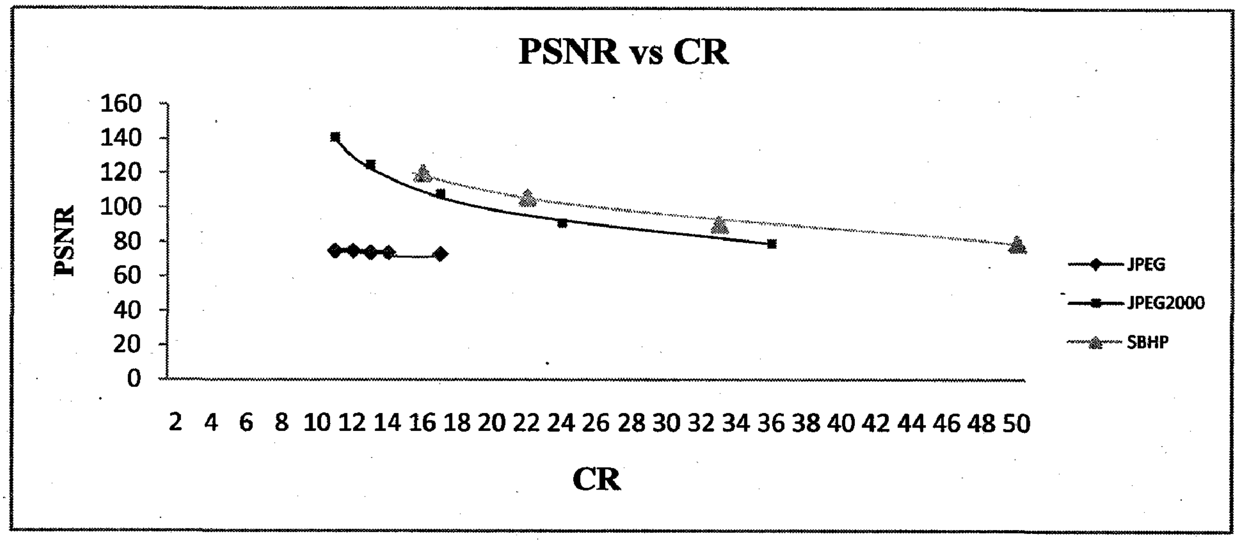


Fig.6.93 A graph between PSNR vs CR

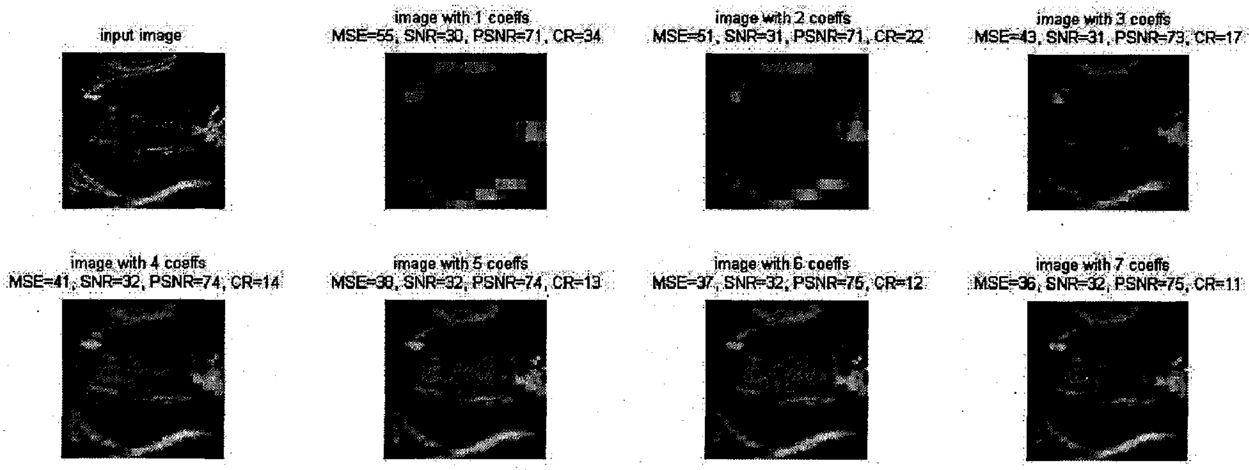


Fig.6.94 Output of JPEG

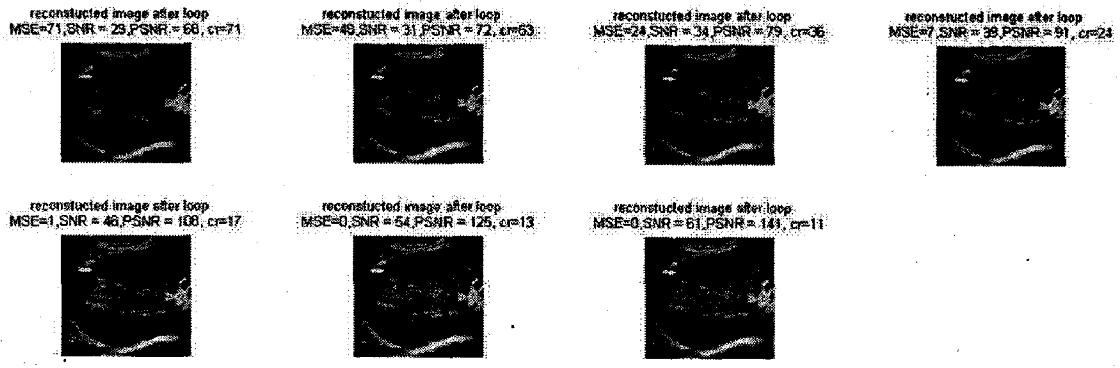


Fig.6.95 Output of JPEG2000

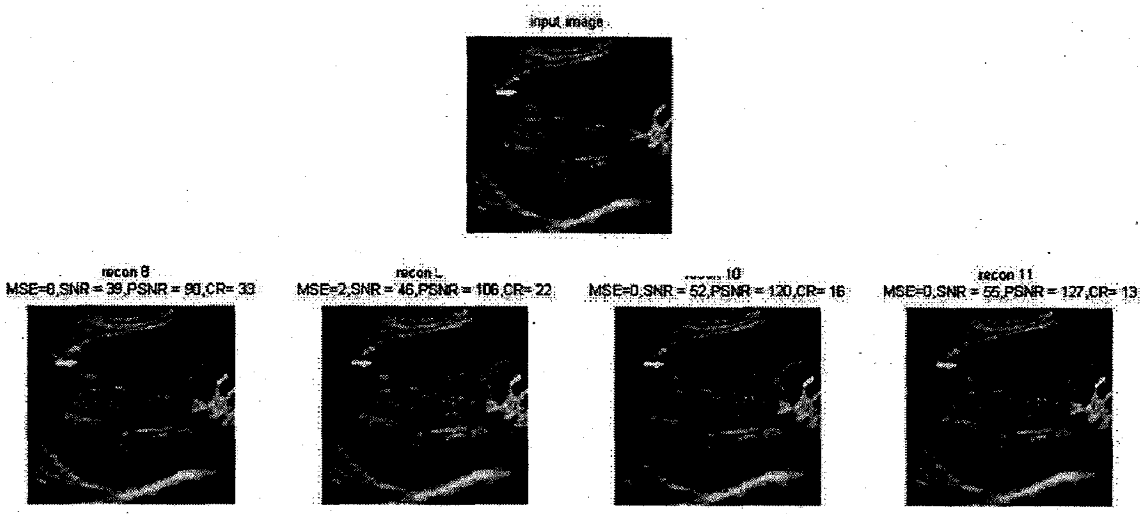


Fig.6.96 Output of SBHP

Image7:

Table6.49.JPEG Output for kidney image7:

coefficients	Mse	snr	Psnr	cr
5	21	34	80	14
6	20	35	81	13
7	19	35	82	12
8	17	35	83	11
9	15	36	84	10

Table6.50.JPEG2000 Output for kidney image7:

loop	Mse	snr	Psnr	cr
6	16	36	83	43
7	5	40	94	30
8	1	47	110	21
9	0	54	126	16
10	0	61	142	12

Table6.51.SBHP Output for kidney image7:

loop	Mse	snr	Psnr	cr
7	16	35	83	81
8	6	40	93	50
9	1	47	109	30
10	0	52	122	21
11	0	55	129	16

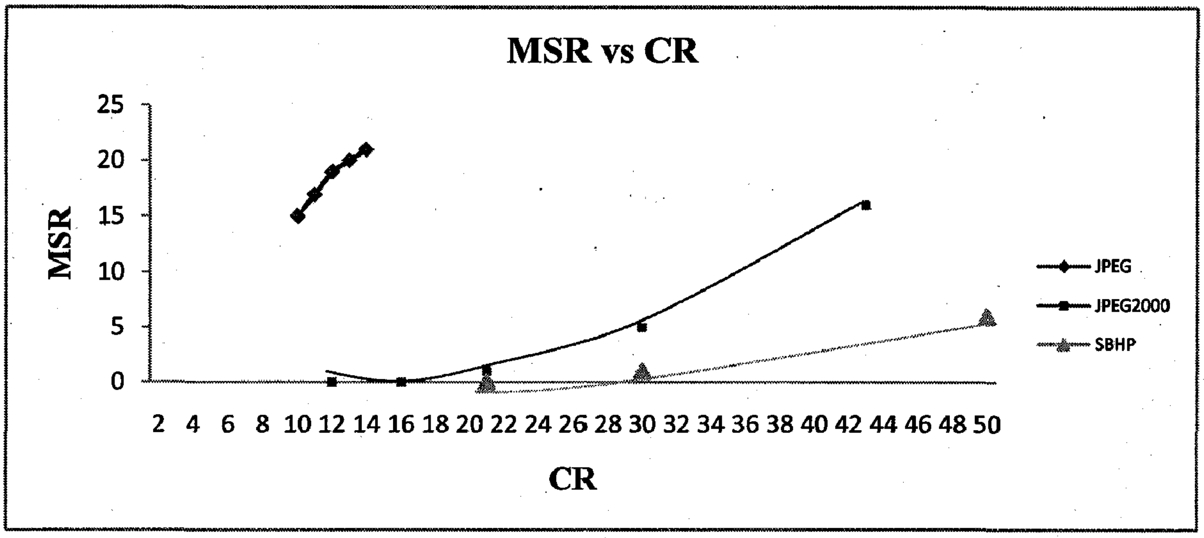


Fig.6.97 A graph between MSR vs CR

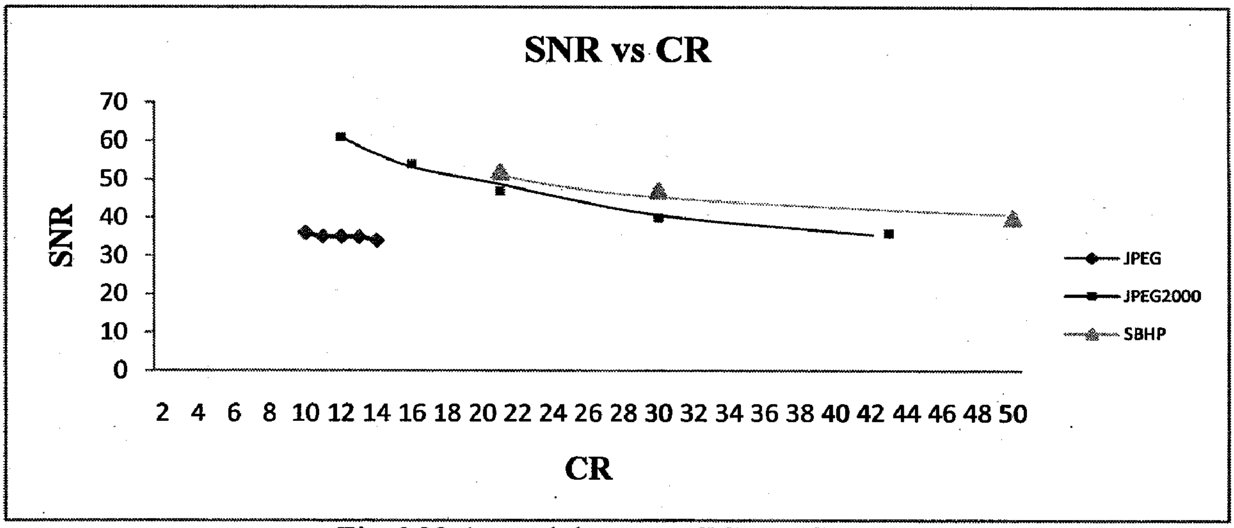


Fig.6.98 A graph between SNR vs CR

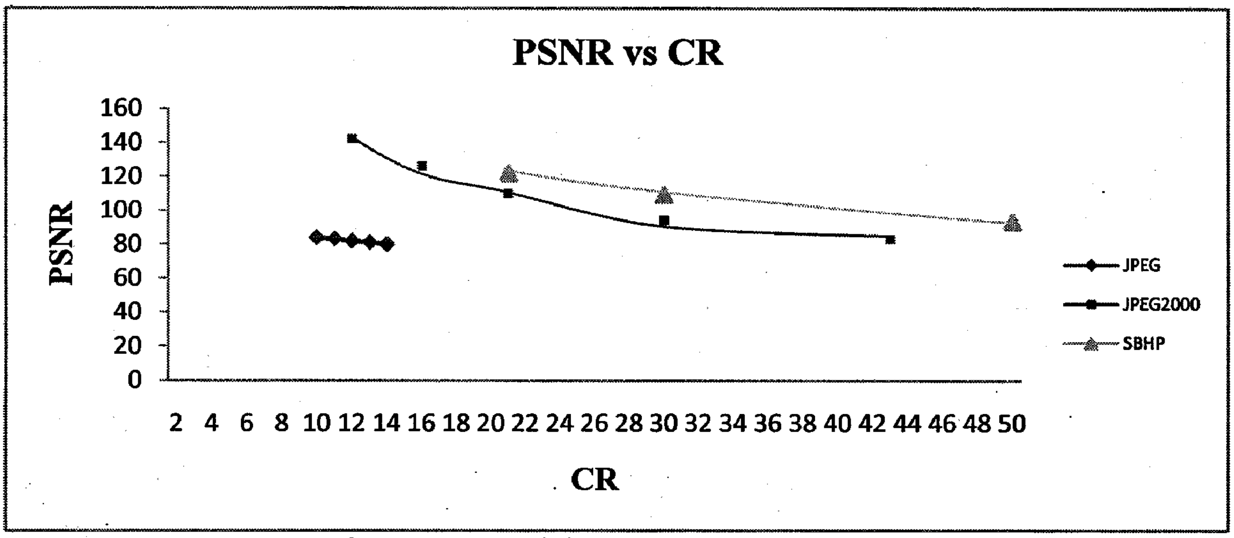


Fig.6.99 A graph between PSNR vs CR

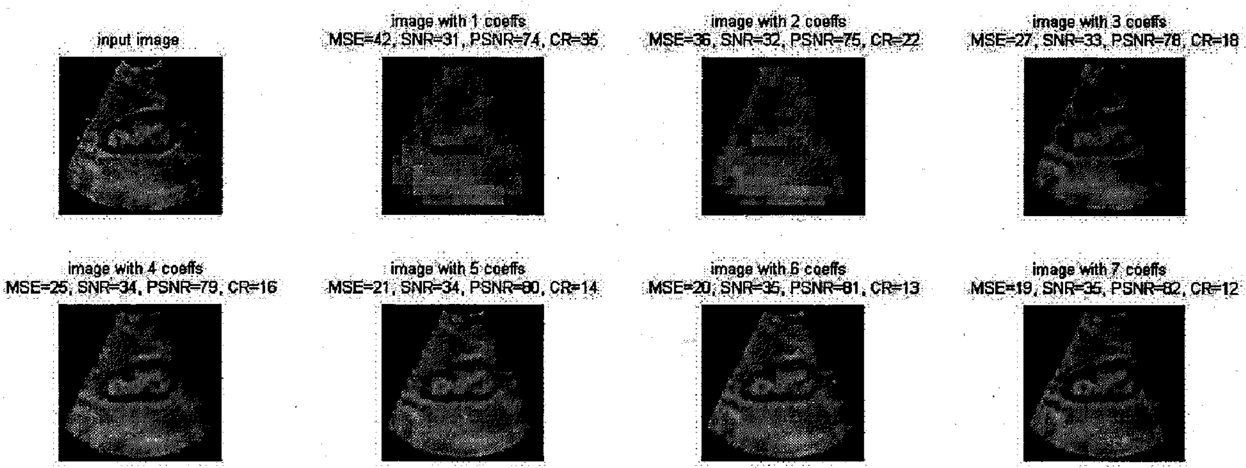


Fig.6.100 Output of JPEG

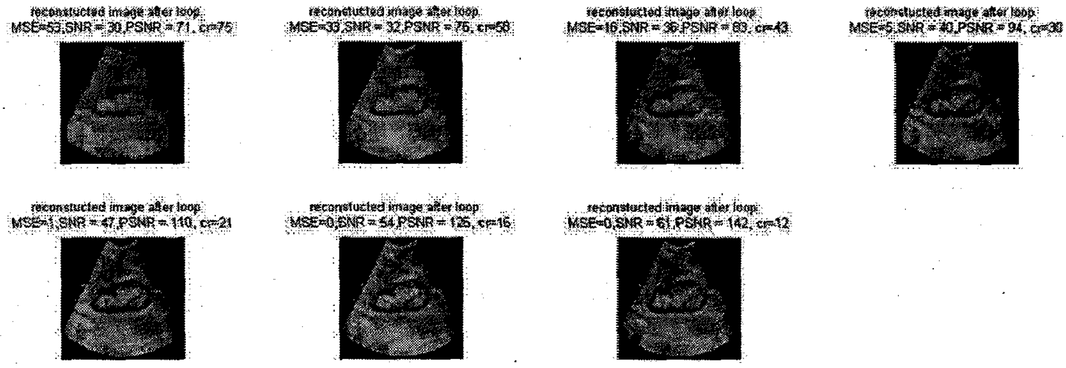


Fig.6.101 Output of JPEG2000

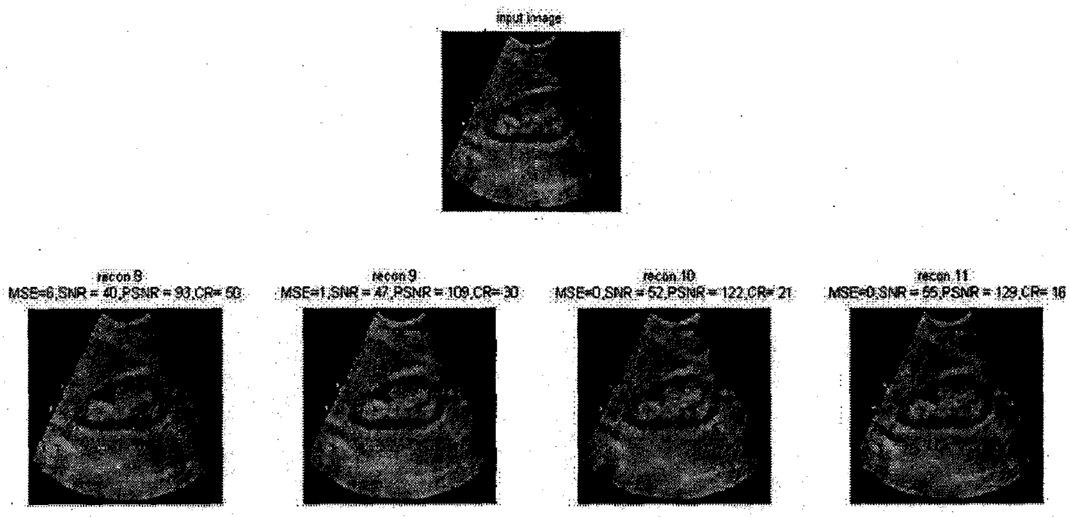


Fig.6.102 Output of SBHP

Image8:

Table6.52.JPEG Output for kidney image8:

coefficients	Mse	snr	Psnr	cr
4	27	33	78	14
5	22	34	80	13
6	19	35	82	12
7	17	35	82	11
8	14	36	84	10

Table6.53.JPEG2000 Output for kidney image8:

loop	Mse	snr	Psnr	cr
6	18	35	82	42
7	6	40	93	29
8	1	47	109	21
9	0	54	125	16
10	0	61	141	12

Table6.54.SBHP Output for kidney image8:

loop	Mse	snr	Psnr	cr
7	18	35	82	80
8	6	40	93	49
9	1	46	108	30
10	0	52	121	21
11	0	55	127	15

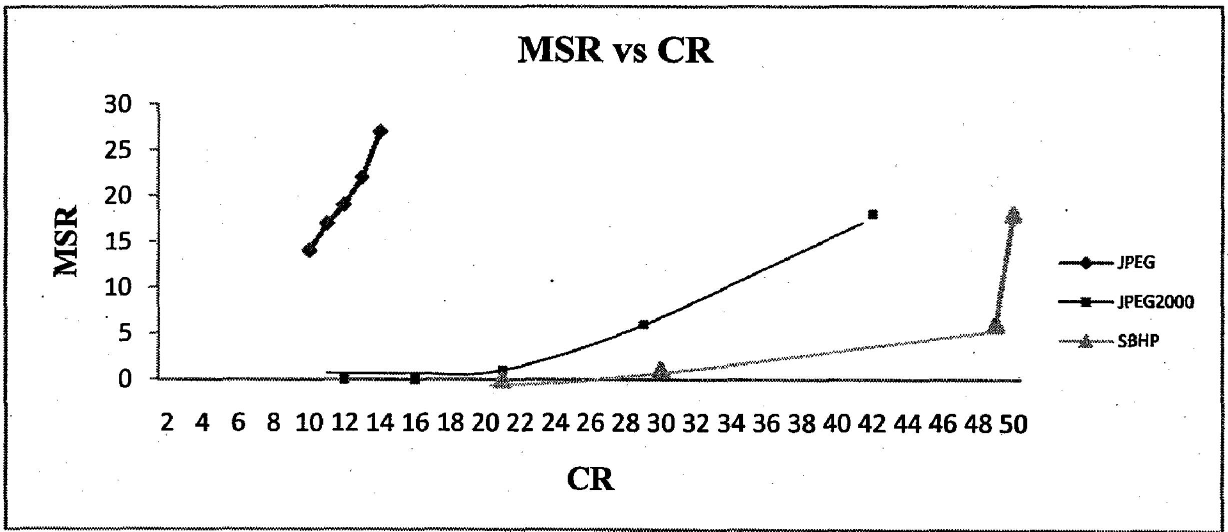


Fig.6.103 A graph between MSR vs CR

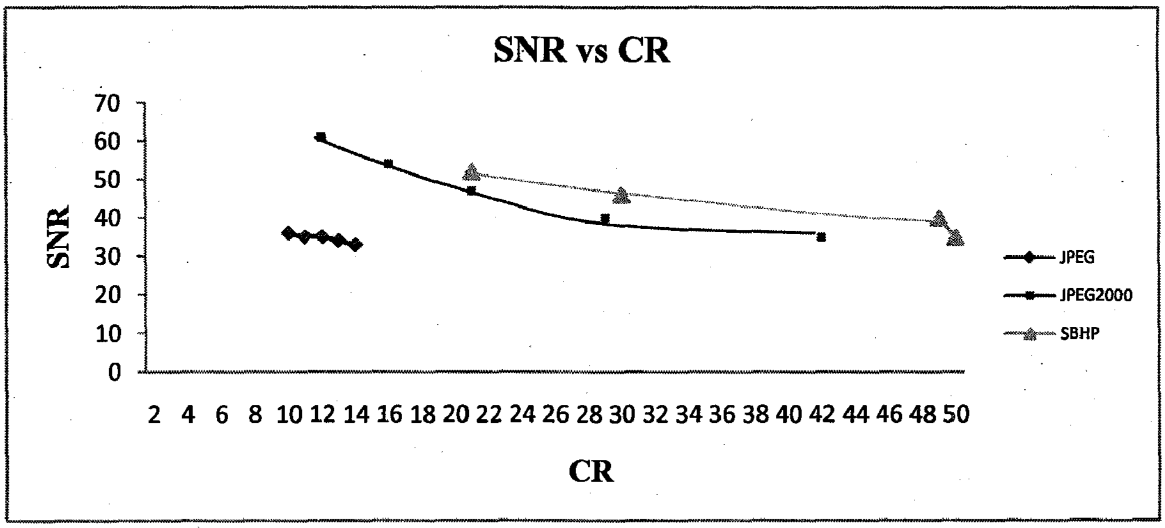


Fig.6.104 A graph Between SNR vs CR

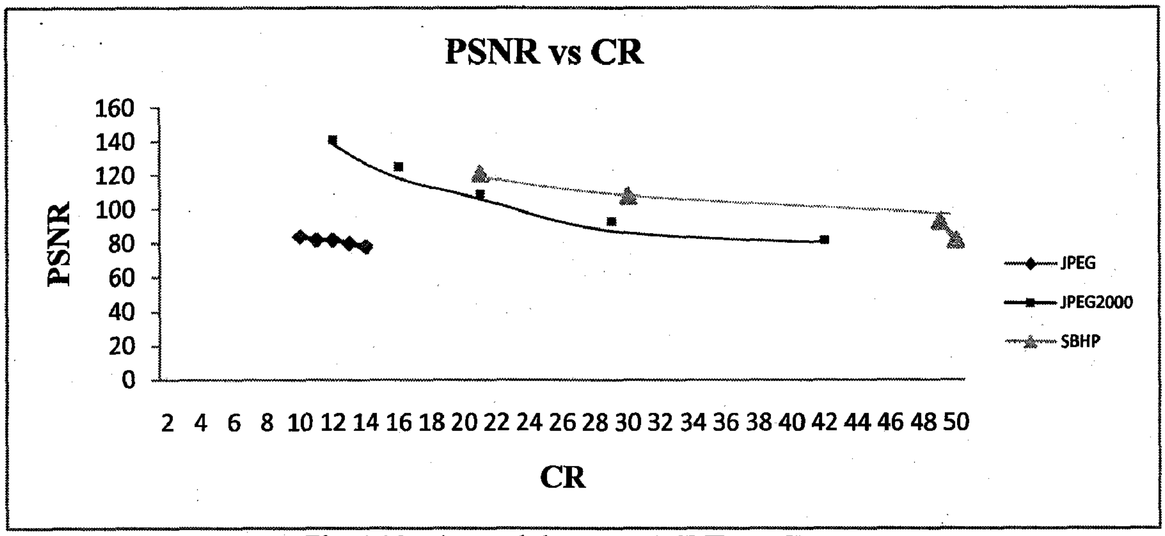


Fig.6.105 A graph between PSNR vs CR

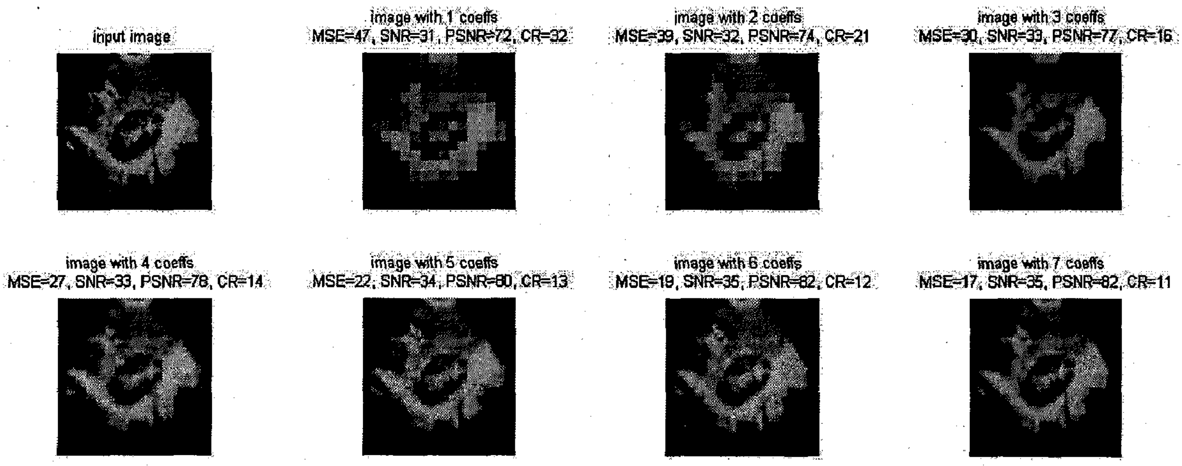


Fig.6.106 Output of JPEG

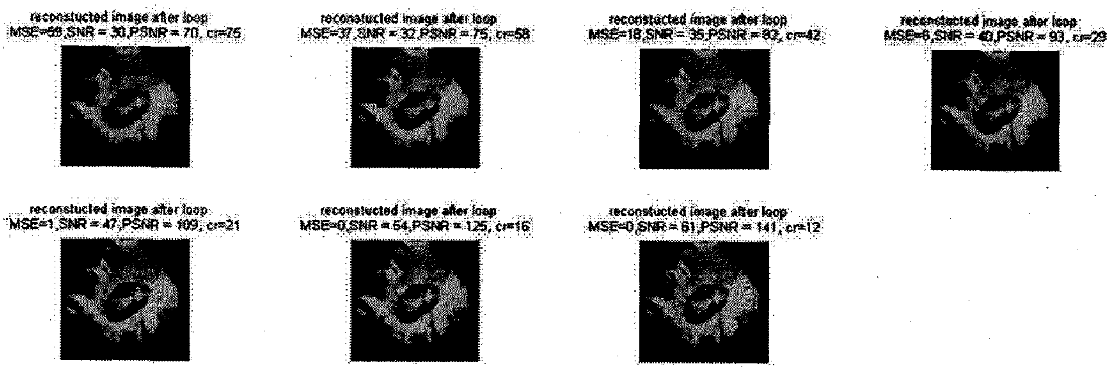


Fig.6.107 Output of JPEG2000

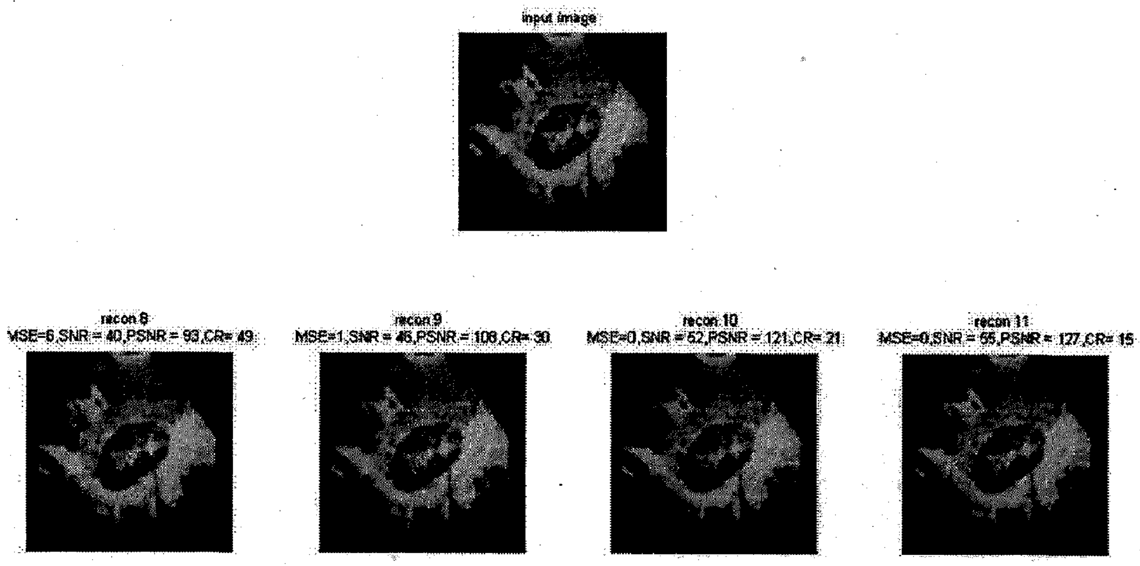


Fig 6.108Output of SBHP

Image9:

Table6.55.JPEG Output for kidney image9:

coefficients	Mse	snr	Psnr	cr
5	28	33	78	14
6	26	33	78	13
7	26	34	78	12
8	25	34	79	11
9	22	34	80	10

Table6.56.JPEG2000 Output for kidney image9:

loop	Mse	snr	Psnr	cr
6	20	35	81	38
7	6	40	92	26
8	1	47	109	19
9	0	54	126	14
10	0	62	143	12

Table6.57.SBHP Output for kidney image9:

loop	Mse	snr	Psnr	cr
7	21	34	80	64
8	7	39	92	40
9	1	46	107	25
10	0	53	122	19
11	0	55	129	15

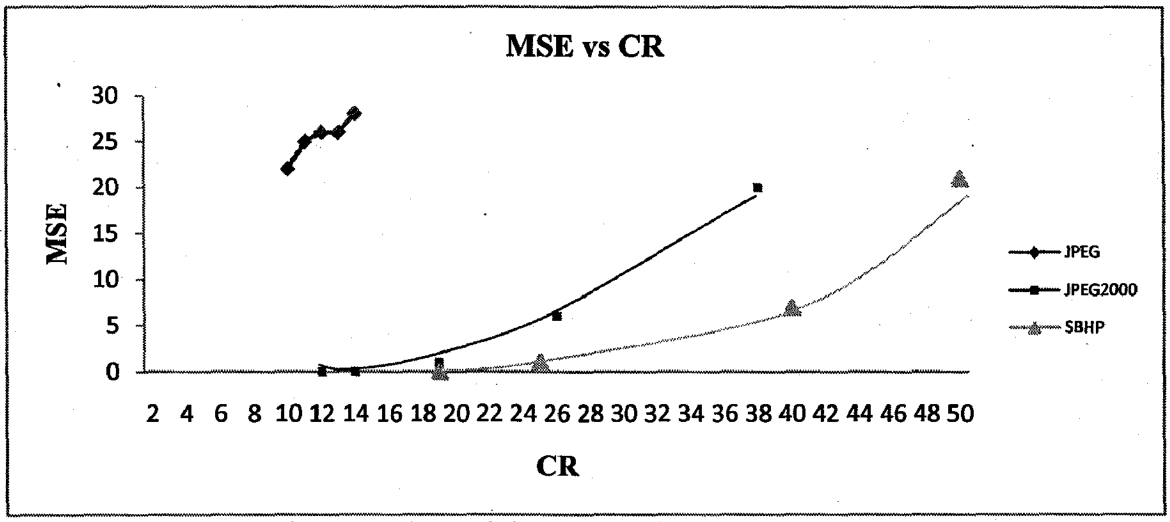


Fig.6.109 A graph between MSE vs CR

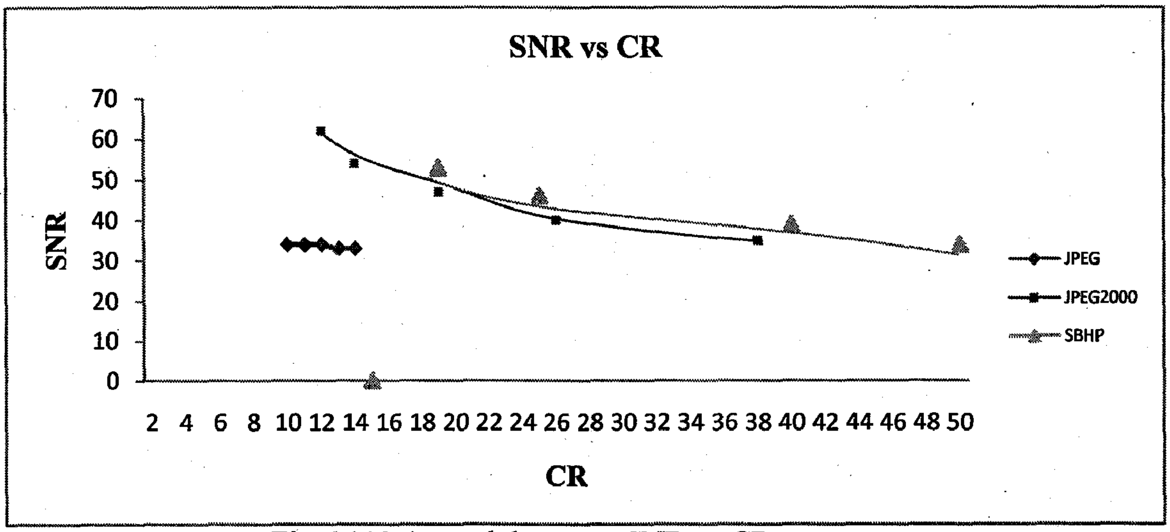


Fig.6.110 A graph between SNR vs CR

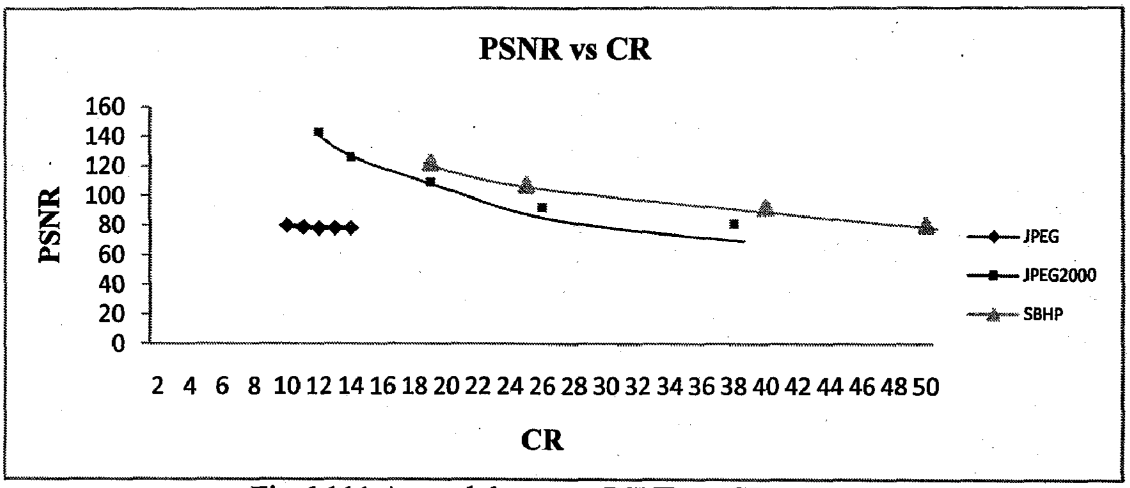
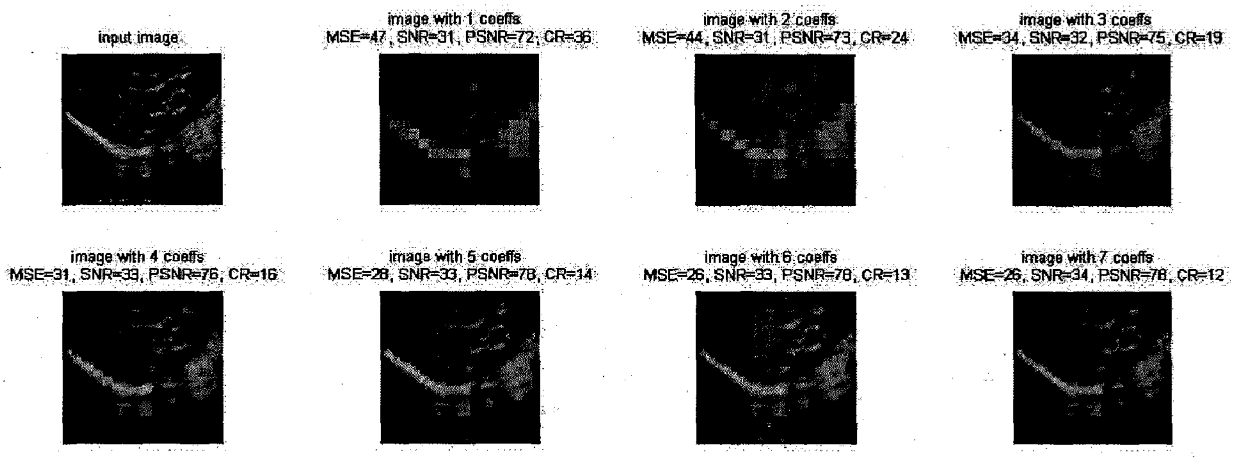


Fig.6.111 A graph between PSNR vs CR



Fif.6.112 Output of JPEG

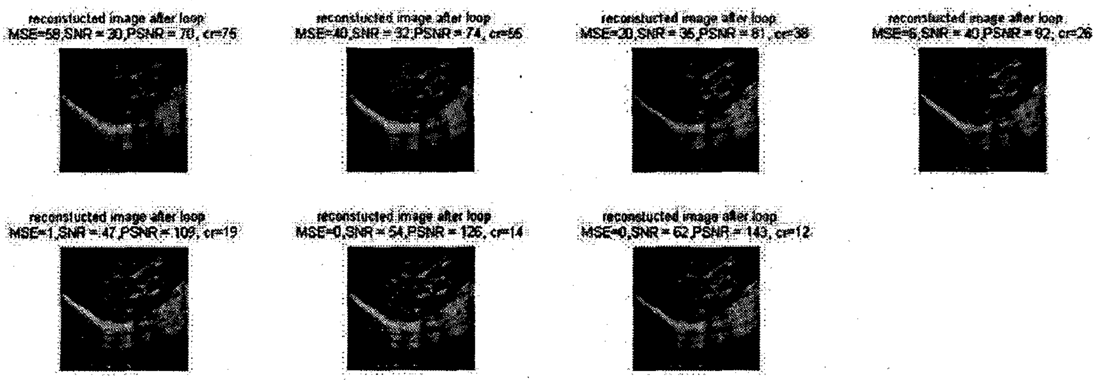


Fig.6.113 Output of JPEG2000

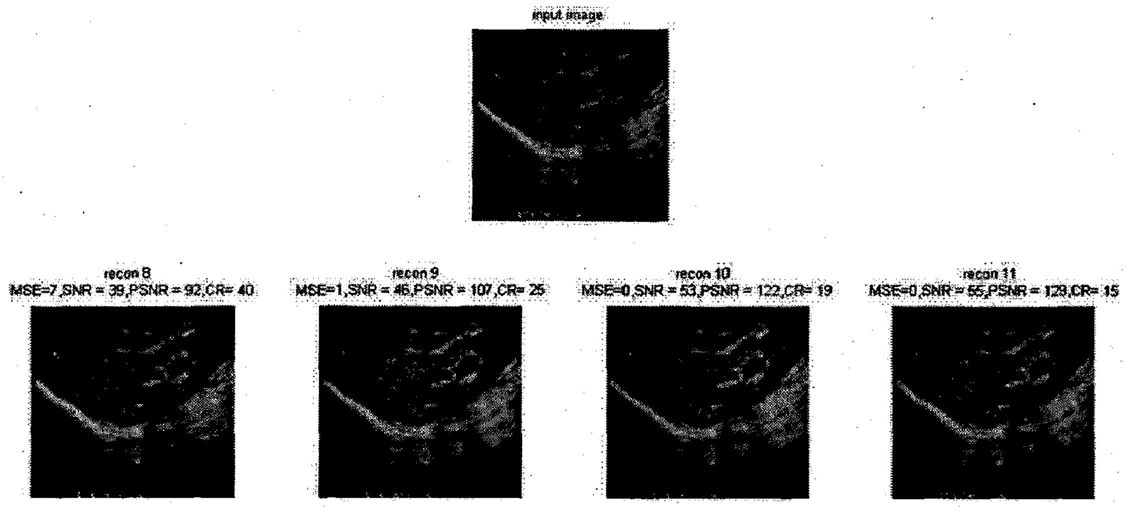


Fig.6.114 Output of SBHP

Image10:

Table6.58.JPEG Output for kidney image10:

coefficients	Mse	snr	Psnr	cr
7	30	33	77	14
8	29	33	77	13
9	28	33	77	12
10	27	33	78	11
11	25	34	79	10

Table6.59.JPEG2000 Output for kidney image10:

loop	Mse	snr	Psnr	cr
6	6	40	92	30
7	1	47	108	21
8	0	54	125	15
9	0	61	142	12
10	0	62	143	12

Table6.60.SBHP Output for kidney image10:

loop	Mse	snr	Psnr	cr
7	7	39	92	43
8	1	46	107	26
9	0	52	121	18
10	0	55	127	14
11	0	55	129	15

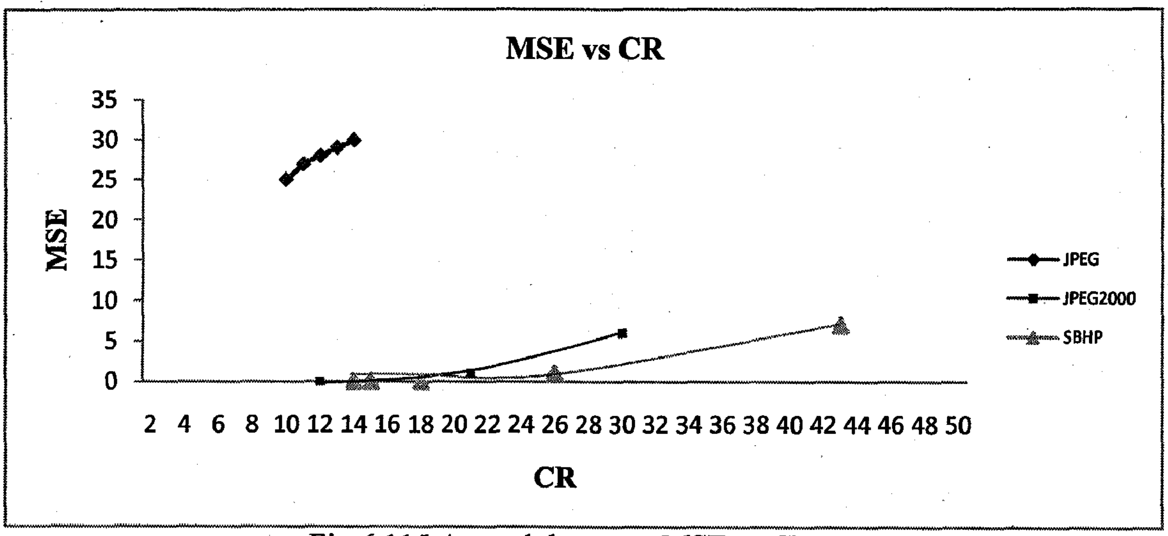


Fig.6.115 A graph between MSE vs CR

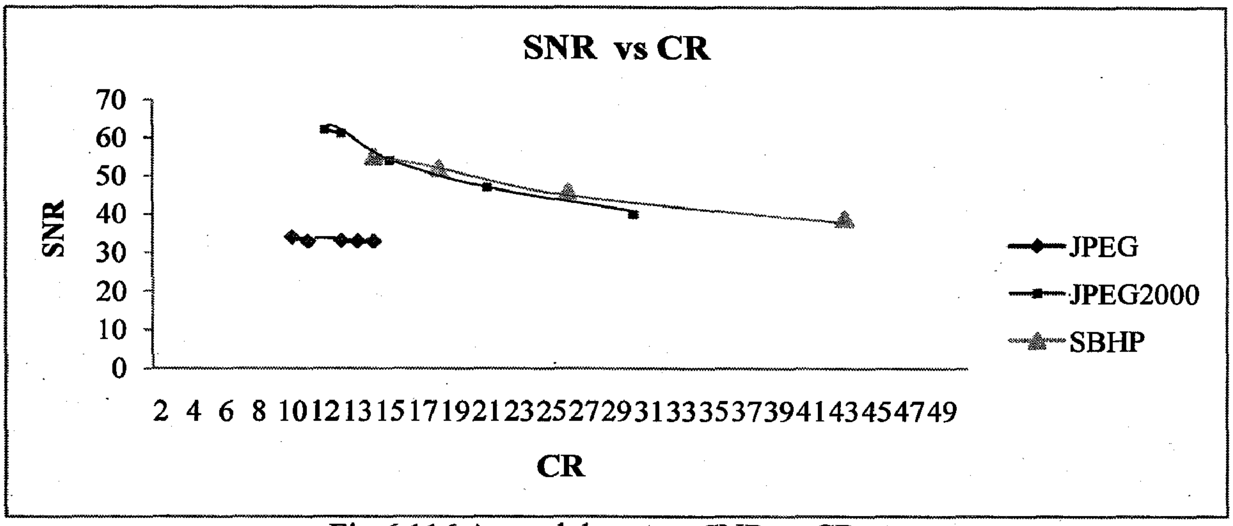


Fig.6.116 A graph between SNR vs CR

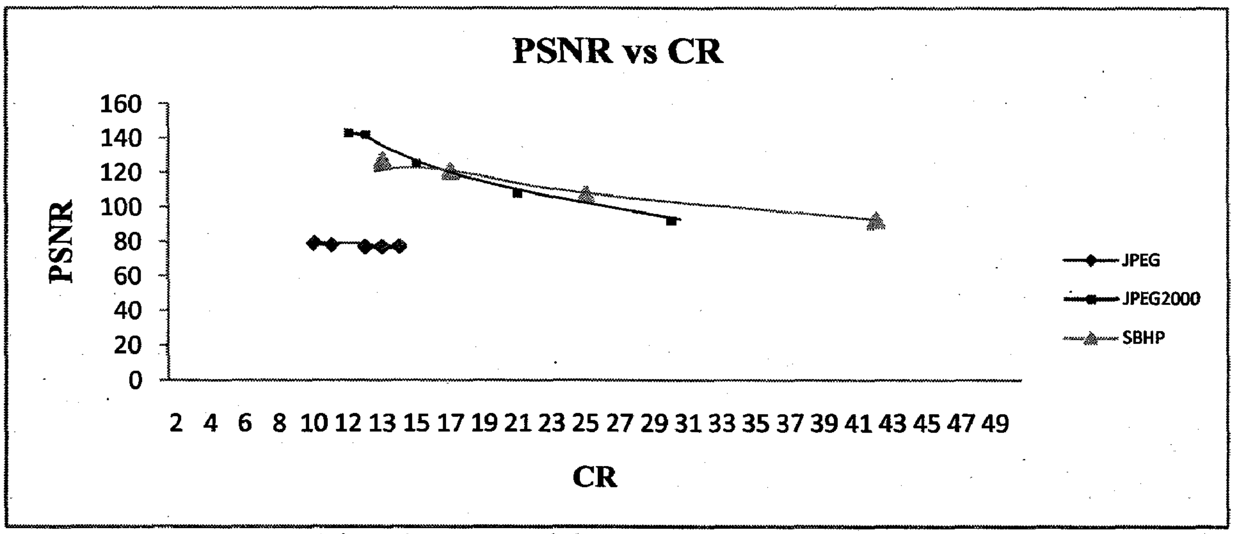


Fig.6.117 A graph between PSNR vs CR

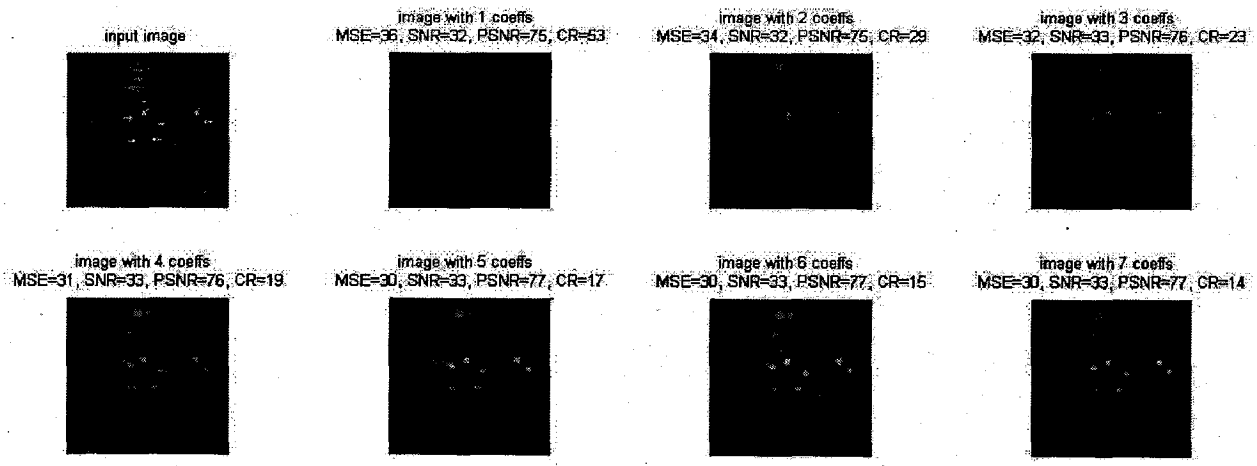


Fig.6.118 Output of JPEG

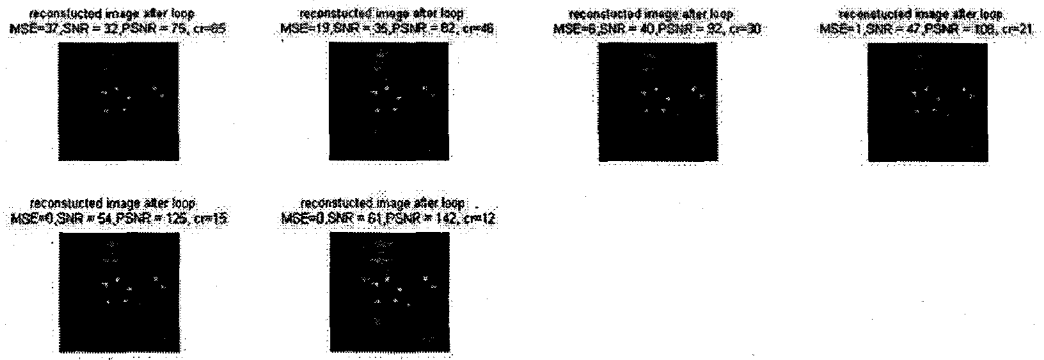


Fig.6.119 Output of JPEG2000

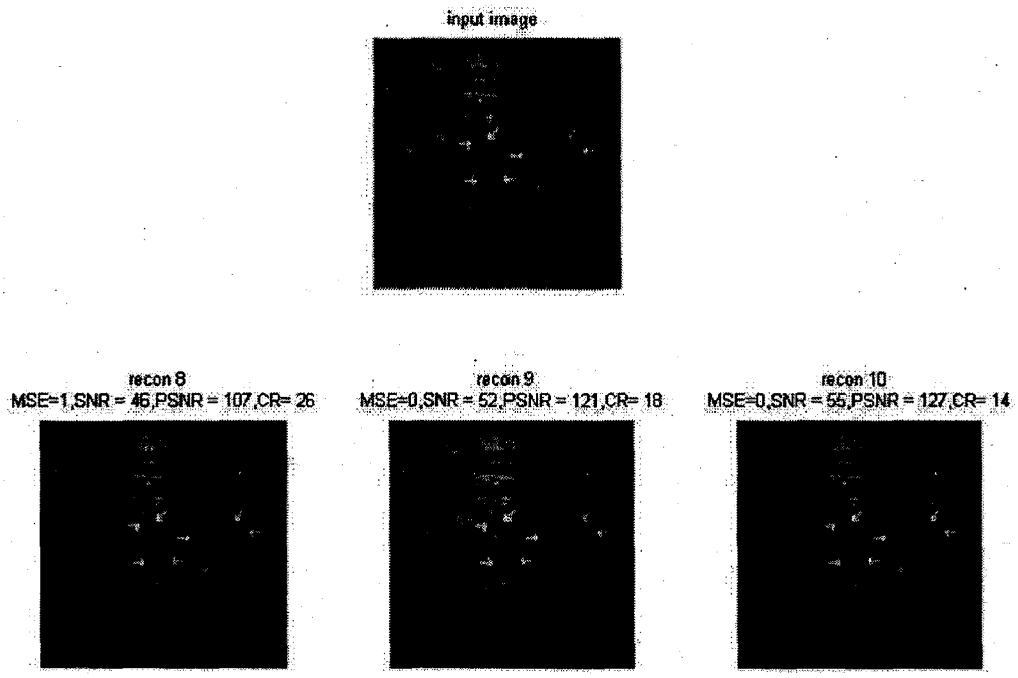


Fig.6.120 Output of SBHP

From the 3 graphs of each image it is observed that,

MSE is low in case of SBHP method as compared to JPEG & JPEG2000.

SNR is high in case of SBHP method as compared to JPEG & JPEG2000.

PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

Chapter 7
CONCLUSIONS AND FUTURE SCOPE

7.1 Conclusions:

The easy, rapid, and reliable digital transmission and storage of medical images would be a tremendous boon to the practice of medicine. Patients readmitted to hospitals could have earlier imaging studies instantly available. Rather than waiting for others to finish with hardcopy films, medical and surgical teams collaborating on patient care could have simultaneous access to imaging studies on monitors throughout the hospital. This long-term digital archiving or rapid transmission is prohibitive without the use of image compression to reduce the file sizes. In this thesis three methods (JPEG, JPEG2000, and SBHP) are implemented on ultrasound images.

The JPEG algorithm yields good results for compression ratios of 10:1 and below (on 8-bit gray-scale images), but at higher compression ratios the underlying block nature of the transform begins to show through the compressed image. By the time compression ratios have reached 24:1, only the DC (lowest frequency) coefficient is getting any bits allocated to it, and the input image has been approximated by a set of 8×8 blocks. Consequently, the decompressed image has substantial blocking artifacts for medium and high compression ratios. This disadvantage is eliminated by JPEG 2000. JPEG 2000 has many advantages over JPEG, such as better image quality at the same file size, 25-35% smaller file sizes at comparable image quality, good image quality even at very high compression ratios (over 80:1), and progressive rendering and transmission through a layered image file structure.

The main disadvantage with JPEG2000 is that it does not consider the energy concentration in the same level and also complexity is more. This is done by SUBBAND HIERARCHICAL BLOCK PARTIONING (SBHP) algorithm. It exploits two fundamental characteristics of an image transform—the well defined hierarchical structure, and energy clustering in frequency and in space and encodes with very low complexity.

In this thesis I developed MATLAB programs for image compression using JPEG, JPEG2000 and SBHP.I applied these three methods on ultrasound images. I applied the three methods (JPEG, JPEG2000, SBHP) on 10 liver ultrasound images and 10 kidney ultrasound images, and I calculated and tabulated MSE, SNR, PSNR and CR for 10 Liver and 10 Kidney ultrasound images. I drew the graph between Mse vs Cr, SNR vs CR, and PSNR vs CR.I also given the output reconstructed images for every input image. From the graphs, I observed that in case of MSE vs CR graphs SBHP method gives better Mse as compared to JPEG & JPEG2000 methods for a given Compression Ratio (CR). In case of SNR vs CR graphs SBHP method gives better SNR as compared to JPEG & JPEG2000 methods for a given Compression Ratio (CR).). In case of PSNR vs CR graphs SBHP method gives better PSNR as compared to JPEG & JPEG2000 methods for a given Compression Ratio (CR).

1. MSE is low in case of SBHP method as compared to JPEG & JPEG2000.
2. SNR is high in case of SBHP method as compared to JPEG & JPEG2000.
3. PSNR is high in case of SBHP method as compared to JPEG & JPEG2000.

7.2. FUTURE SCOPE:

1. In the present work the compression methods are designed only for grey scale images. But in practice true color medical images are also available. Hence the present work can be extended for true color images.
2. In the present work, I am not considering region of interest (ROI).Hence this work can be extended to ROI compression.

REFERENCES

1. Jayant, N., Johnston, J., Safranek,R, "Signal Compression Based On Models Of Human Perception", *Proc. of the IEEE*, 1993, Vol. 81, pp. 1385-1422.
2. Zovko-Cihlar, B., Grgic, S. and Modric, D, "Coding Techniques In Multimedia Communications", *Proceedings of the 2nd International Workshop on Image and Signal Processing, IWISP'95*, Budapest, 08-10 Nov. 1995, pp. 24-32.
3. *ISO/IEC IS 10918 (1991)*, "Digital Compression and Coding of Continuous Tone Still Images", 1991.
4. D.Wu and E.C.Tan,"Comparison of loss less image compression". TENCON 99 *proceedings of the IEEE region 10 conference*, Cheju island, South korea, vol.1, 1999, pages 718-721.
5. M. H. El Ayadi M. M. Syiam A. A. Gamgoum" A Comparative study of lossless image compression techniques" *IJICIS*, Vol. 5, No. 1, July 2005.
6. Bauer, S., Zovko-Cihlar, B. and Grgic, M. "The Influence of Impairments from Digital Compression of Video Signal on Perceived Picture Quality", *Proceedings of the 3rd International Workshop on Image and Signal Processing, IWISP'96*, Manchester, 04-07 Nov. 1996, pp. 245-248.
7. Lewis, A.S.and Knowles, G., "Image Compression Using the 2-D Wavelet Transform", *IEEE Trans. On Image Processing*, 1992, Vol. 1, pp. 244-250.
8. Antonini, M., Barland, M., Mathieu, P. and Daubechies, I., "Image Coding Using the Wavelet Transform", *IEEE Trans. On Image Processing*, 1992, Vol. 1, pp. 205-220.
9. Grgic, S., Kers, K. and Grgic, M., "Image Compression Using Wavelets", *proceedings of the IEEE International Symposium on International Electronics, ISIE'99*, 12-16 Jul 1999, Vol. 1, pp. 99-104.
10. Amir Said "A New, fast, and efficient image codec based on set partitioning in hierarchical trees", *IEEE transactions on circuits and systems for video technology*, June 1996, vol. 6, no. 3, pp. 243 – 250.
11. Charilaos Christopoulos" The JPEG2000 still image coding system: an overview" *IEEE Transactions on consumer electronics*, November 2000, Vol. 46, No. 4, pages 1103-1127

12. Chaur-Chin Chen, "On the selection of image compression algorithms," *14th International conference on pattern recognition (ICPR'98),* 1998, Volume 2.
13. Amhamed saffor, Abdul Rahman Ramli, Kwan-Hoong Ng, "A Comparative study of Image Compression between JPEG and WAVELET". *Malaysian journal of computer science*, June 2001, vol. 14 no. 1, Pages.39-45.
14. Mr.T.Sreenivasulu reddy, Ms. K. Ramani, Dr. S. Varadarajan, Dr. B.C.Jinaga "Image compression using transform coding methods" ,*International journal of Computer science and network security*, July 2007, vol.7, no.7, pp. 51-56.
15. Andrew B. Watson" Image compression using the Discrete Cosine Transform" *Mathematica journal*, 1994, Vol. 4, no.1, pp. 81-88.
16. Gregory K. Wallace" The JPEG still picture compression standard" *IEEE Transactions on consumer electronics*, Feb. 1992, Vol. 38, no.1, pp. 18-34.
17. Recommendation T.81 , "Information technology digital compression and coding of continuous-tone still images requirements and guidelines" *The international telegraph and telephone consultative committee, T.81*,November 1992
18. G. K. Kharate, A. A. Ghatol and P.P. Rege, "Image Compression Using Wavelet Packet Tree", *ICGST-GVIP Journal*, July 2005, Vol. (5), no. 7, pp. 37-40.
19. Michael W. Marcellin, Michael J. Gormish, Ali Bilgin, Martin P. Boliek", An Overview of JPEG-2000", *Proc. of IEEE data compression conference*, 28-30 March 2000, pp. 523 - 541
20. JIN LI," Image compression: The mathematics of JPEG 2000", *Modern signal processing MSRI Publications*, 2003, Vol. 46
21. Robert Buckley" JPEG 2000 - a practical digital preservation standard?" *DPC Technology watch series report 08-01*, February 2008.
22. Hong man, "Performance Analysis of the JPEG 2000 Image Coding Standard", *Multimedia Tools and Applications*, 2005, vol.26, pp. 27–57.
23. William A. Pearlman, Asad Islam, Nithin Nagaraj, and Amir Said ,"Efficient, Low-Complexity Image Coding With a Set-Partitioning Embedded Block Coder", *IEEE Transactions On Circuits And Systems For Video Technology,* Nov 2004,Vol. 14, No. 11, pp.1219- 1235.

24. R.Sudhakar, Ms R Karthiga, S.Jayaraman, "Image Compression using Coding of Wavelet Coefficients – A Survey", *ICGST-GVIP Journal*, June 2005,Volume (5), Issue (6), pp.31-32.