M.K.S.S.S's Cummins College of Engineering for Women
Department of Electronics & Telecommunication Engineering
Course: Digital Signal Processing (EC3201)
A.Y. 2019-2020 SEM. II
PROJECT BASED LEARNING
Project Title: IMAGE COMPRESSION AND DECOMPRESSION MODEL

Presented by:
GROUP No.16

Project Group Members
3223 - Trupti Devidas .Jagtap
3217 -Sakshi Sangram .Gite
3211-Aishvarya Sanjay .Dhone
3271-Anuja Anil .Tidke

T.Y.(E&TC) Div. B

Guided by:

Dr. Ashwini .Deshpande

E&TC Department, CCEW, Pune

PHASE- I EVALUATION
MARCH- WEEK 1

COURSE: DSP (PROJECT BASED LEARNING)

Contents

- Work Plan
- Task list and distribution
- Objectives
- Literature Survey
- System Block Diagram

Work Plan

Month & Week no. / Task / Module	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6
1. Literature Survey & Concept development						
2. System-level design (Block Schematic)						
3. Detail design (Algorithm, Design Parameters,Software Programming) for Compression and Decompression						
4. Testing of Software Module (Simulation Result)						
5. Testing of the Complete System						
6. Internal Project Demo.						
7. Project Report						

Task List and Distribution

Trupti Devidas .Jagtap:	Algorithm Design for Compression,Testing for Decompression
Sakshi Sangram .Gite:	Algorithm Design for Decompression,Testing for Compression
Aishvarya Sanjay .Dhone:	Code for Decompression,Testing for Compression
Anuja Anil .Tidke:	Code for Compression,Testing for Compression

Objectives

- To study different transform that can be applied to compress and decompress an Image.
- Detailed study of Discrete Cosine Transform(DCT).
- To code a user defined function to Compress and Decompress Image using DCT.

Literature Survey

- Data compression is the process of reducing data required to represent given amount of information
- The property of the DCT that makes it quite suitable for compression is its high degree of "spectral compaction;" at a qualitative level, a signal's DCT representation tends to have more of its energy concentrated in a small number of coefficients when compared to other transforms like the DFT.
- The property of the DCT that makes it quite suitable for compression is its high degree of "spectral compaction;" at a qualitative level, a signal's DCT representation tends to have more of its energy concentrated in a small number of coefficients when compared to other transforms like the DFT.
- Image Compression plays a major role in many important and diverse applications including Televideo conferencing, Remote sensing, Document ,Medical imaging ,Fax.

[Reference: Digital Image Processing, by Rafael C.Gonzalez and Richard E.Woods]

Image as a Signal

The amount of light reflected by an object in the physical world (3d world) is pass through the lens of the camera and it becomes a 2d signal and hence result in image formation. It is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically.The value of f(x,y) at any point is gives the pixel value at that point of an image.This image is then digitized using methods of signal processing and then this digital image is manipulated in digital image processing.

LOSSY	LOSSLESS
Reconstructed data after compression different from original data.	Reconstructed data exactly like original.
Used to compress multimedia data (audio, video, still images).	preferred for text and data files, such as bank records, text articles, etc.
suffer from generation loss.	Not affected.
More compression possible.	Lossless data compression must always make some files *longer.*
For eg: JPEG	For eg: GIF,TIFF,PNG

References:Websites;[1]https://www.geeksforgeeks.org/difference-between-lossy-compression-and-lossless-compression/
[2]https://www.tutorialspoint.com/difference-between-lossy-compression-and-lossless-compression

Types of Transforms

- DFT
- DCT
- Hadamard
- Walsh
- Wavelet
- Haar
- KL

WHY DCT ?

➢ The property of the DCT that makes it quite suitable for compression is its high degree of "spectral compaction;" at a qualitative level, a signal's DCT representation tends to have more of its energy concentrated in a small number of coefficients when compared to other transforms like the DFT.

References Papers:[1]Survey paper on image compression techniques Sudha Rawat1, Ajeet Kumar Verma2 1,2M.tech Babasaheb bhimrao ambedkar university Department of Computer Science, Babasaheb bhimrao university, Lucknow, U.P
[2]The Role of Transforms in Image Compression V. K. Bairagi ,A. M. Sapkal ,M. S. Gaikwad
[3]https://www.math.cuhk.edu.hk/~lmlui/dct.pdf
Website:[1]http://user.engineering.uiowa.edu/~dip/lecture/DataCompression.html

DCT Matrix

The N*N Cosine Transform matrix is

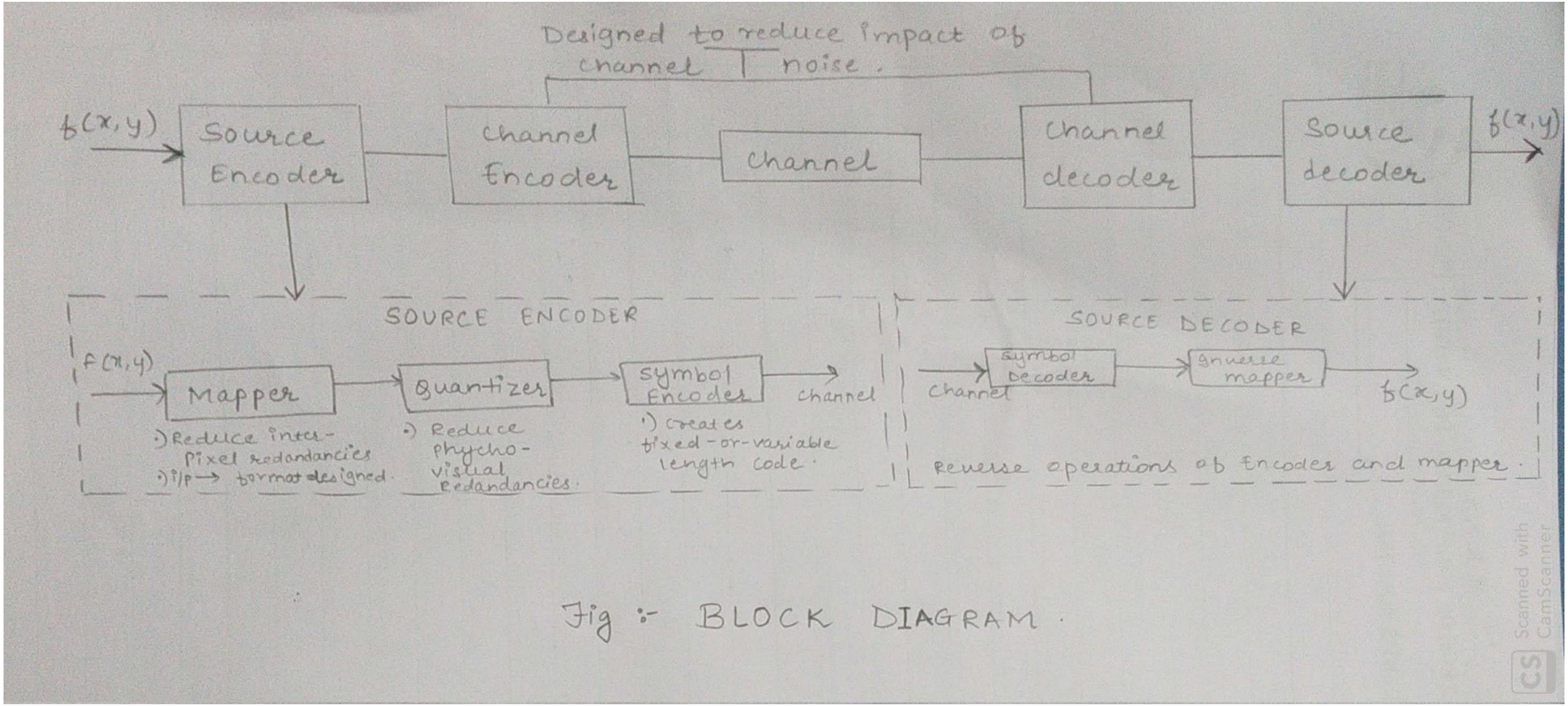
$$X[k] = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos[(2n+1)\Pi k/2N]$$

for $0 <= k <= N-1$

$$\alpha(k) = \sqrt{1/N} \quad \text{for } k = 0$$

$$\alpha(k) = \sqrt{2/N} \quad \text{for } k \neq 0$$

References:[1]https://www.math.cuhk.edu.hk/~lmlui/dct.pdf
[2]Digital Image Processing, by Rafael C.Gonzalez and Richard E.Woods



BLOCK DIAGRAM

Reference:[1]Digital Image Processing, by Rafael C.Gonzalez and Richard E.Woods

PHASE- II EVALUATION
MARCH- WEEK 4

COURSE: DSP (PROJECT BASED LEARNING)

Methodology

1) The discrete cosine transform (**DCT**) represents an image as a sum of sinusoids of varying magnitudes and frequencies. The DCT has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT.

2)DCT transformation is used due to its energy compaction characteristics.

3)DCT has cosine function which is easier to compute and the number of coefficients become less. Thus, DCT results into more accurate image reconstruction .

There are both lossy and lossless image compression format available and Lossy compression is the most popular image compression on JPEG images

4)DCT and DSP:

Discrete cosine transform is the fundamental part of JPEG compression and one of the most widely used conversion technique in digital signal processing (**DSP**) and image compression.

Due to the importance of the discrete cosine transform in JPEG standard, an algorithm is proposed that is used for compression and decompression of image using DCT.

Step ① :- Consider 8 x 8 block of an image.

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 166 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ & 65 & 60 & 70 & 79 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

The range of intensities — 0 to 255

Step ② :- changing range from −128 to 127

Subtracting 128 from each pixel,

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Step ③ :- $X[k] = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)\pi k}{2N}\right]$

$\alpha(k) = \sqrt{1/8}$ for $k = 0$.

$\alpha(k) = \sqrt{2/8}$ for $k \neq 0$.

Step ④ :- Quantization matrix,

$$Q_{j,k} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$B_{j,k} = \text{round}\left(\frac{A_{j,k}}{Q_{j,k}}\right)$$

$$B_{j,k} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This is the compressed Image.

Applying DCT to obtain the Compressed Image

Detail Design

1)JPEG compression takes place in five steps - color space conversion, down sampling, discrete cosine transformation (DCT),quantization and entropy encoding.

2)**COMPRESSION:**

We are converting the image to YCbCr for the ease of operation on the chrominance and luminance component of the image.The chrominance components are down sampled to half of its size as changes in the chrominance are not identified easily, exploiting the fact that human eye is more sensitive to intensity changes ,rather than color changes.

It is apparent that the total number of bits present in an image can be minimized by removing the redundant bits.

There are three types of redundancy available in terms of space, time and spectrum.

- Spatial redundancy indicates the correlation between neighboring pixel values.
- Spectral redundancy indicates correlation among different color planes.
- Temporal redundancy indicates correlation among different frames in an image. Compression techniques or methods aim to reduce the spatial and spectral redundancy with maximum efficiency.

```
DCT COEFFICENTS for [8 8] at compression of 10

dct_coeff =

    0.7789    0.0047   -0.0023    0.0033   -0.0025    0.0016   -0.0017    0.0004
    0.0047   -0.0025   -0.0005   -0.0013    0.0006   -0.0000    0.0008    0.0002
   -0.0023   -0.0005    0.0032   -0.0012    0.0015   -0.0016    0.0003   -0.0008
    0.0033   -0.0013   -0.0012   -0.0004   -0.0001    0.0005    0.0004    0.0003
   -0.0025    0.0006    0.0015   -0.0001    0.0005   -0.0007   -0.0001   -0.0004
    0.0016   -0.0000   -0.0016    0.0005   -0.0007    0.0008   -0.0001    0.0004
   -0.0017    0.0008    0.0003    0.0004   -0.0001   -0.0001   -0.0003   -0.0001
    0.0004    0.0002   -0.0008    0.0003   -0.0004    0.0004   -0.0001    0.0002
```

3)**DECOMPRESSION:**

Decompression is the reverse process of compression.
Firstly, the decoder takes the compressed image data as its input.
It then applies a run length decoding, inverse zigzag, dequantization, inverse discrete cosine transform (IDCT), then obtains the reconstructed image.

NOTE: In digital image processing the YCbCr color space is often used in order to take advantage of the lower resolution capability of the human visual system for color with respect to luminosity. Thus, RGB to YCbCr conversion is used in image processing

Algorithm [1,3]

Steps for Compression:

1. Read the image from the user and convert it to Gray/YCbCr
2. Level off the image ,by subtracting 128 from the original image
3. Consider a block size of 8 for segmentation
4. Brake the levelled off image into 8x8 blocks of matrix
5. Write a function to calculate the DCT Transform matrix; the function is written in accordance:

The N*N Cosine Transform matrix is

$$X[k] = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos[(2n+1)\Pi k/2N]$$

$$\text{for } 0 <= k <= N-1$$

$$\alpha(k) = \sqrt{1/N} \quad \text{for } k = 0$$

$$\alpha(k) = \sqrt{2/N} \quad \text{for } k \neq 0$$

6. Now the DCT is applied to the each Block by multiplying the block ,with a DCT on left and the transpose of DCT on its right
7. Each block is Quantized using a Quantization matrix and setting a quality/compression level(1-100)

   The quantization matrix is as follows:

```
DCT_quantizer = [ 16 11 10 16 24 40 51 61;
                  12 12 14 19 26 58 60 55;
                  14 13 16 24 40 57 69 56;
                  14 17 22 29 51 87 80 62;
                  18 22 37 56 68 109 103 77;
                  24 35 55 64 81 104 113 92;
                  49 64 78 87 103 121 120 101;
                  72 92 95 98 112 100 103 99 ];
```

Quantization is achieved by dividing each element in the- transformed image matrix in step 6 by corresponding element in the quantization matrix, and then rounding to the nearest integer value

8. The array of compressed blocks that constitute the quantized values ,is stored in a final matrix

Steps for De-compression:

9. Reconstruction of our image begins by multiplying each element of Quantized matrix by the corresponding element of the quantization matrix originally used
10. The IDCT is applied to the resulted matrix by multiplying the resultantant matrix of step 9 with transpose of DCT on left and DCT on right
11. Final 128 is added to the each element of the matrix of step 10 to produce the decompressed ,original image
12. Calculating PSNR and Plotting the magnitude spectrum of original image and compressed image(through DCT and through DFT)

PHASE- III EVALUATION
APRIL-WEEK 2

COURSE: DSP (PROJECT BASED LEARNING)

Software Used:Matlab

- Reading the image from
- Writing functions for Compression,Quantization and Decompression of the image using blockprop()
- Plotting Magnitude Spectrums of Original and Reconstructed image using fftshift(),fft(),abs()

Program

CODE FOR DCT COMPRESSION

```
clear all;
close all;
quant_multiple = input('add the level of compression');%accepting level of compression from user
folder='MATLAB Drive/*.jpeg';%add the path here
ext='*.jpeg' ; %the image extension
[filename, path] = uigetfile(fullfile(folder,ext)) ;%opens a dialog box that lists files in the current folder
original_image = imread(fullfile(path, filename));%reading the image
imshow(original_image);%display the image
display('Compressing the image...');
display('Please wait...');
original_image=rgb2ycbcr(original_image);%converting image to gray scale image
original_image=original_image(:,:, 1);
```

continued...

```
% ------------or replace line 4 and 5 by --------------
% original_image=rgb2gray(original_image);
% -----------------------------------------------------
original_image=original_image-128;%level off by 128


blocksize=8;%taking a blocksize of 8
resized_image=imresize(original_image,[512 512]);%resizing the image
resized_image=im2double(resized_image);%making the values double


%-----------------Function for DCT MATRIX--------------%
i = 0;
for j = 0:blocksize-1
DCT_transpose(i+1,j+1)=sqrt(1/ blocksize)*cos((2*j+1)*i*pi/(2*blocksize));
end
```

continued...

```
for i = 1:blocksize-1
for j = 0:blocksize-1
DCT_transpose(i+1,j+1) = sqrt(2/blocksize)*cos((2*j+1)*i*pi/(2*blocksize));
end
end
%------------------Function for DCT MATRIX ENDS---------%



%-------------------Step 6------------------------------%
dct_matrix = @(block_struct) DCT_transpose * block_struct.data * DCT_transpose';%D=T*(Image)*T; function to
compress the image .Create block processing function.
resultant_dct_matrix = blockproc(resized_image,[8 8],dct_matrix);%applying above function for 8x8 matrix of
resized image and store the result in resultant_matrix
%Process the image, block-by-block.
%--------------------Step 6 ends-------------------------%

%-----------------Quantization of image-----------------%
```

continued...

```
%-----------------Quantization of image--------------------%
quant_multiple=0.05*10/quant_multiple;
DCT_quantizer = [ 16 11 10 16 24 40 51 61;%quantization matrix
12 12 14 19 26 58 60 55;
14 13 16 24 40 57 69 56;
14 17 22 29 51 87 80 62;
18 22 37 56 68 109 103 77;
24 35 55 64 81 104 113 92;
49 64 78 87 103 121 120 101;
72 92 95 98 112 100 103 99 ];
```

```
quant_func= @(block_struct)((block_struct.data) ./ (DCT_quantizer* quant_multiple)+0.5)%Declaring
Quantization function
quantized_image = blockproc(resultant_dct_matrix,[8 8],quant_func)%Applying above function for each 8x8
matrix of resultant_dct_matrix and store result in quantization_image
quantized_image = floor(quantized_image);%applying func floor (to round off to the lowest) to quantized_image
```

continued...

```
figure,imshow(quantized_image);%displaying quantized_image
imwrite(quantized_image,' Compressed.jpeg');%storing the quantized_image in Compressed.jpeg
display('Compression completed!!!');

%------------------End of Quantization------------------%



%-------------------Compression is Done-----------------------%




%-----------------------Decompression of Image-------------------%
display('Decompressing the image for you...');
display('Please wait....');
reconstructed_matrix= blockproc(quantized_image,[8 8],@(block_struct)DCT_quantizer.*block_struct.data);
%reconstruction of matrix
```

continued...

```
inverse_dct = @(block_struct)ceil(DCT_transpose' * block_struct.data * DCT_transpose);   %Inverse DCT
inv_D=T'*(Image)*T
decompressed_image = blockproc(reconstructed_matrix,[8 8],inverse_dct);%applying the above function for 8x8 to
reconstructed_mattrix
figure, imagesc(decompressed_image), colormap gray;
imwrite(decompressed_image,'Decompressed Image.jpeg');%storing the decompressed image in Decompressed.jpeg
display('Compression and Decompression Successfull');
display('Please check the "Quantization.jpeg" & "Decompressed.jpeg" in your folder');


%Plot 2D DFT magnitude spectrum of original signal
fs = 1000;
t = 0:1/fs:1-(1/fs);
X1 = fftshift(fft(original_image));
```

continued...

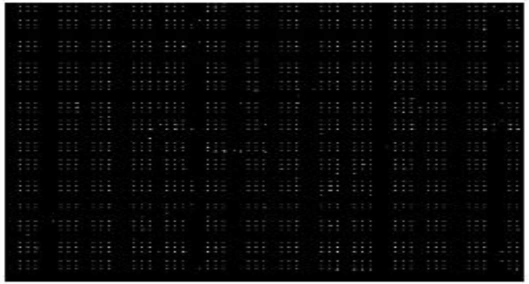
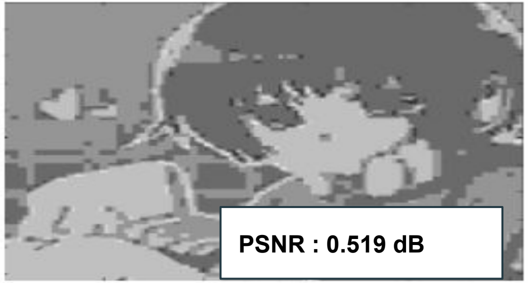
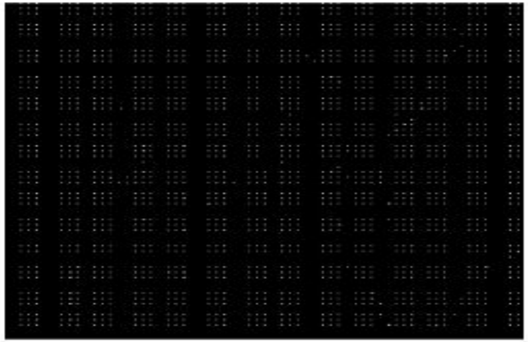
```
X1 = abs(X1);
figure,plot(X1/fs);
title('Magnitude Spectrum of original signal');
X2=fftshift(fft(decompressed_image));
X2=abs(X2);
figure ,plot(X2/fs);
title('Magnitude Spectrum of reconstructed image signal');
fprintf(1, 'Finding the signal-to-noise ratio...\n');   %calculating the PSNR
PSNR=0;
for row=1:rows
 for col=1:cols
   PSNR=PSNR+(resized_image(row,col)-decompressed_image(row,col))^2;
 end
end
X=(255^2)/(1/(((rows+cols)/2)^2)*PSNR);
PSNR=log10(X);
fprintf(1,'\nThe signal-to-noise ratio (PSNR) for one half of the coefficient to be zero is: %1.3f dB\n\n',
PSNR);
```

Simulation Results

With Q Factor of 100(High Quality Image Obtained)

ORIGINAL IMAGE	COMPRESSED IMAGE	DECOMPRESSED IMAGE
		 PSNR: 0.587 dB
		

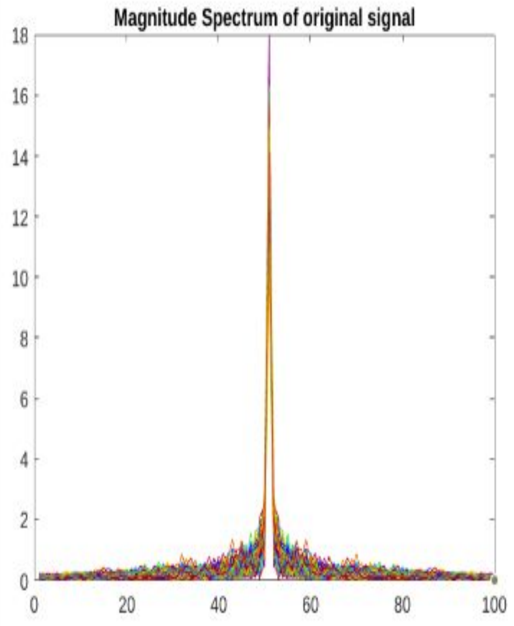
With Q-Factor of 10(Low Quality Image Obtained)

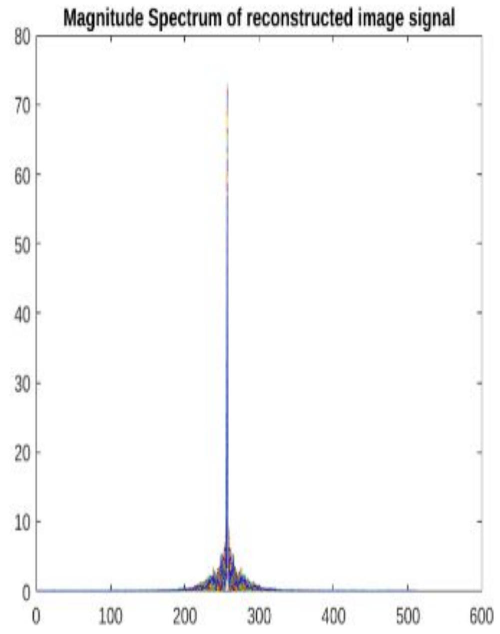
ORIGINAL IMAGE	COMPRESSED IMAGE	DECOMPRESSED IMAGE
		 PSNR : 0.519 dB
		

System Analysis

Comparing DCT and DFT

DCT COMPRESSION

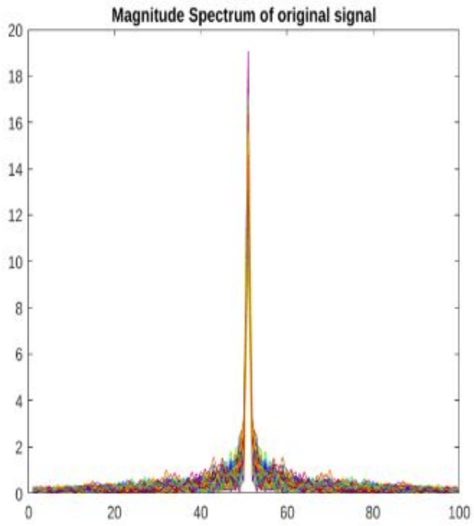




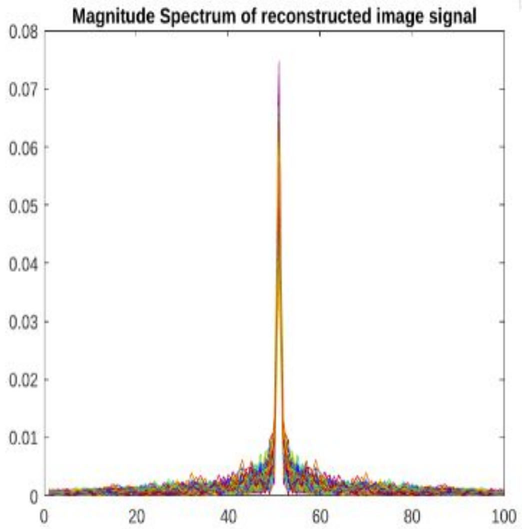
Comparing DCT and DFT

DFT COMPRESSION

MAGNITUDE SPECTRUM OF ORIGINAL IMAGE



MAGNITUDE SPECTRUM OF DECOMPRESSED IMAGE



- In DCT most of the signal energy is concentrated in only a small number of coefficients as compared to DFT.
- DFT requires large number of multiplications and additions for the calculation.Thus, DFT is taking long time for computation as compared to DCT.
- In computing the DFT of a signal, there is the implicit extension of several copies of the signal placed one after the other (n-point periodicity). The resultant discontinuities require several frequencies for good representation. As against this, the discontinuities are reduced in a DCT because a reflected copy of the signal is appended to it (2n-point periodicity).Thus,DCT has better energy compaction than DFT. [2]
- Pixel blocks in DFT are strictly periodic ,thus possibly producing large discontinuities in the gray values at the edge.
- Pixel blocks in DCT are reversed , replicating gray values across the edges of blocks and reducing the discontinuities at the edge.

Applications

Following are the applications in which these compression techniques can be used.

1. E-Health Systems
2. Telemedicine
3. Video Conferencing
4. Monitoring and Surveillance

These are the major applications where these compression techniques can be used and the need for using is a
compulsion because remote areas have limited resources, bandwidth limitation is one of the limited resources which
need more financial requirement

Conclusion

To optimally utilize the resources like bandwidth, storage space, etc compression has become a necessity in today's world. This project presents simulation of one such technique called JPEG compression and decompression model using Lossy Algorithm. The fact that human eye is less sensitive to high intensity variation and color variation in an image. Hence, we remove such information and thus leading to compression. The compressed image will contain the same perceived information as the input image. The technique is well explained in the above sections.Simulation Result and System Analysis helps to conclude that the given technique can be useful in image compression without losing the perceived information content of the image.

References:

[1]Image Compression from Digital Image Processing, by Rafael C.Gonzalez and Richard E.Woods
[2]Image Compression and DCT byKeen Cabin and Peter Gent
[3]https://www.geeksforgeeks.org/discrete-cosine-transform-algorithm-program/
[4]https://www.youtube.com/watch?v=sckLJpjH5p8
[5]https://www.youtube.com/watch?v=1BcvE8jAf6I
[6]https://www.researchgate.net/publication/257076110_Image_Compression_Comparison_between_Discrete_Cosine_Transform_and_Fast_Fourier_Transform_and_the_problems_associated_with_DCT