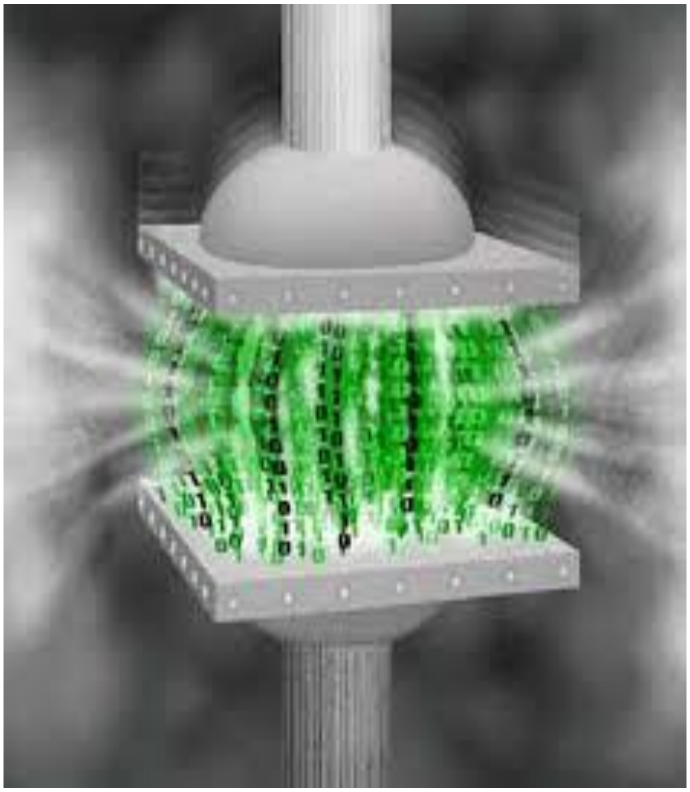


JPEG Compression Standard

-JIGAR GADA

USC ID: 8979-1025-58
jigargada90@gmail.com
Graduate Student, EE Dept.
University of Southern California

LAST UPDATED ON 3/2/2014
Image Reference:
https://www.ibm.com/developerworks/community/blogs/ibmnas/entry/v7ku_compression?lang=en

Problem 3. JPEG

3.1 Motivation

JPEG is the most widely used method for lossy image compression and is the most common format used for storing images in any camera device and for storing and transmitting images on the internet. JPEG is able to achieve a 10:1 compression with very little loss in the perceptible image quality.

JPEF format is simple, easy to understand and implement. Since it is an open source format, it can be used directly without any patent issues which makes JPEG the hot cake for the multimedia industry. The first JPEG format was released in early 1990's after which the format has continuously improved (latest being JPEG 2000) and offers better compression ratios and better quality.

3.2. Approach

JPEG has left no stone unturned in compressing the file. It has extracted the very minute details of human image perception and tried to compress the file in the best possible way.

Here is the block diagram of JPEG Encoder and decoder which explains its working in brief.

Baseline JPEG Encoder and Decoder

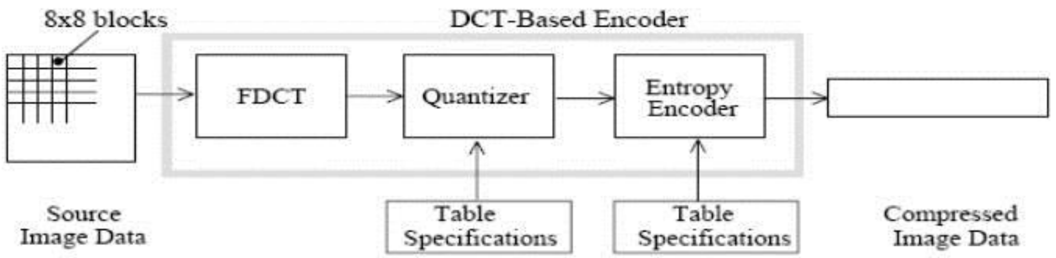


Fig.2 JPEG encoder block diagram [1]

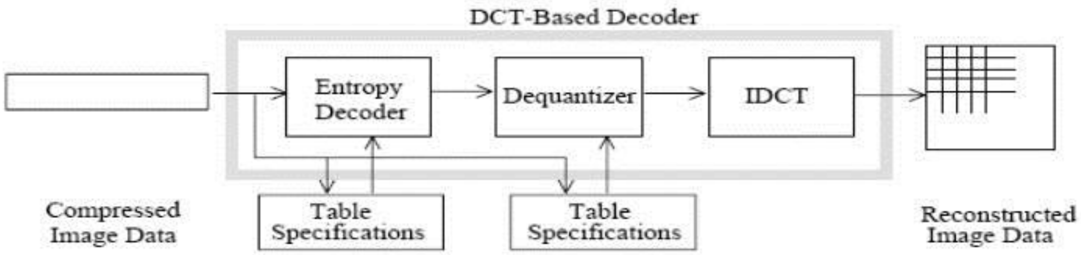


Fig.3 JPEG decoder block diagram [1]

Image Ref: http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/jpeg/jpeg/decoder.htm

The steps involved in JPEG encoding can be split as:

1. Divide the image in 8x8 blocks
2. Take 2-D DCT of each block
3. Quantize the data.
4. Entropy encoding

3.3 Discrete Cosine Transform (DCT) and Quantization for JPEG

To understand the DCT and quantization, the image *lena.raw* having a resolution of 16x16 is provided. All the codes have been written in C++. The steps involved are as follows:

1. **Reading the Gray Scale values of the image**
   Standard IO routine of C++ (*fread*) has been used to read the gray scale values (0-255) of the image. The image read is then converted to 2-d image as rows (16) and columns (16).

2. **Subtract 128 from each pixel value**
   This step is needed because DCT is designed to work on symmetric pixel values ranging from -128 to 127.

3. **DCT of the 8x8 block**

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$C(u) = \begin{cases} 1/\sqrt{2} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases}$$

where *p(x, y)* is the gray scale value of pixel *(x,y)* from part (2), and *D(i,j)* is the DCT coefficient at coordinates *(i,j)*.

We use the above formula to compute the DCT for each block.

4. **Quantize each block with a quality factor Q.**

   A standard quantization matrix Q_{50} with a quality factor of 50 for the JPEG standard is given as follows:

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

The quantized 8x8 matrix C can be obtained by dividing each element in the DCT coefficients matrix D by the corresponding element in the quantization matrix Q, and then rounding to the nearest integer value:

$$C(i,j) = round(\frac{D(i,j)}{Q(i,j)})$$

DCT matrix before quantization

Block 1	**Block 2**

-8 -18 51 12 18 -67 -55 -27
55 -31 51 12 0 28 12 1
-32 78 -25 24 14 -7 12 5
20 2 -18 33 -19 15 -10 4
-4 -9 17 -29 7 -4 1 -6
-15 12 -24 15 -12 -4 0 0
5 -11 8 -13 5 -2 -2 -1
-2 2 0 2 -1 0 0 1
	88 -125 17 -29 -2 -15 -3 3
	-34 17 -109 108 53 -5 21 -13
	-34 -82 5 29 -93 51 17 -25
	53 -45 -50 -73 24 19 -40 21
	32 11 28 -1 23 -51 10 19
	-23 10 49 -13 -6 -19 1 -9
	-2 2 -10 -5 18 11 4 3
	13 5 -4 11 17 0 0 1

Block 3	**Block 4**

58 61 72 -44 -12 122 -47 -37
35 -14 -28 -49 0 20 4 -6
20 -22 60 -31 11 35 -11 -4
-8 11 -2 9 7 0 4 -1
1 12 -16 -3 24 -5 -19 11
0 0 -4 -2 9 -19 4 -1
11 -1 3 11 5 -3 9 -9
-3 2 0 0 3 0 1 0
	89 -4 77 9 -64 -14 0 4
	-7 -97 119 35 1 -67 -6 14
	-23 69 -45 -40 -14 29 50 -43
	-1 28 -30 1 -66 20 1 16
	-43 -19 12 11 19 -3 -36 -15
	0 -33 -5 5 5 0 0 -21
	-1 3 -11 -6 -10 -6 -3 12
	-11 8 0 0 -10 1 -8 0

The quantized values for the *lena.raw* image with quality factor of 50 are as follows:

Block 1	**Block 2**
-1 -2 5 1 1 -2 -1 0	6 12 2 -2 0 0 0 0
5 -3 4 1 0 0 0 0	-3 1 -8 6 2 0 0 0
-2 6 -2 1 0 0 0 0	-2 -6 0 1 -2 1 0 0
1 0 -1 1 0 0 0 0	4 -3 -2 -3 0 0 -1 0
0 0 0 -1 0 0 0 0	2 1 1 0 0 0 0 0
-1 0 0 0 0 0 0 0	-1 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
Block 3	**Block 4**
-12 6 7 -3 -1 -3 -1 -1	6 0 8 1 -3 0 0 0
3 -1 -2 -3 0 0 0 0	-1 -8 9 2 0 -1 0 0
1 -2 4 -1 0 1 0 0	-2 5 -3 -2 0 1 1 -1
-1 1 0 0 0 0 0 0	0 2 -1 0 -1 0 0 0
0 1 0 0 0 0 0 0	-2 -1 0 0 0 0 0 0
0 0 0 0 0 0 0 0	0 -1 0 0 0 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Explanation for the Q matrix

The Q matrix is the only lossy part in JPEG and so it has to be chosen very wisely.

The following image displays the 2-D DCT basis for an 8x8 block

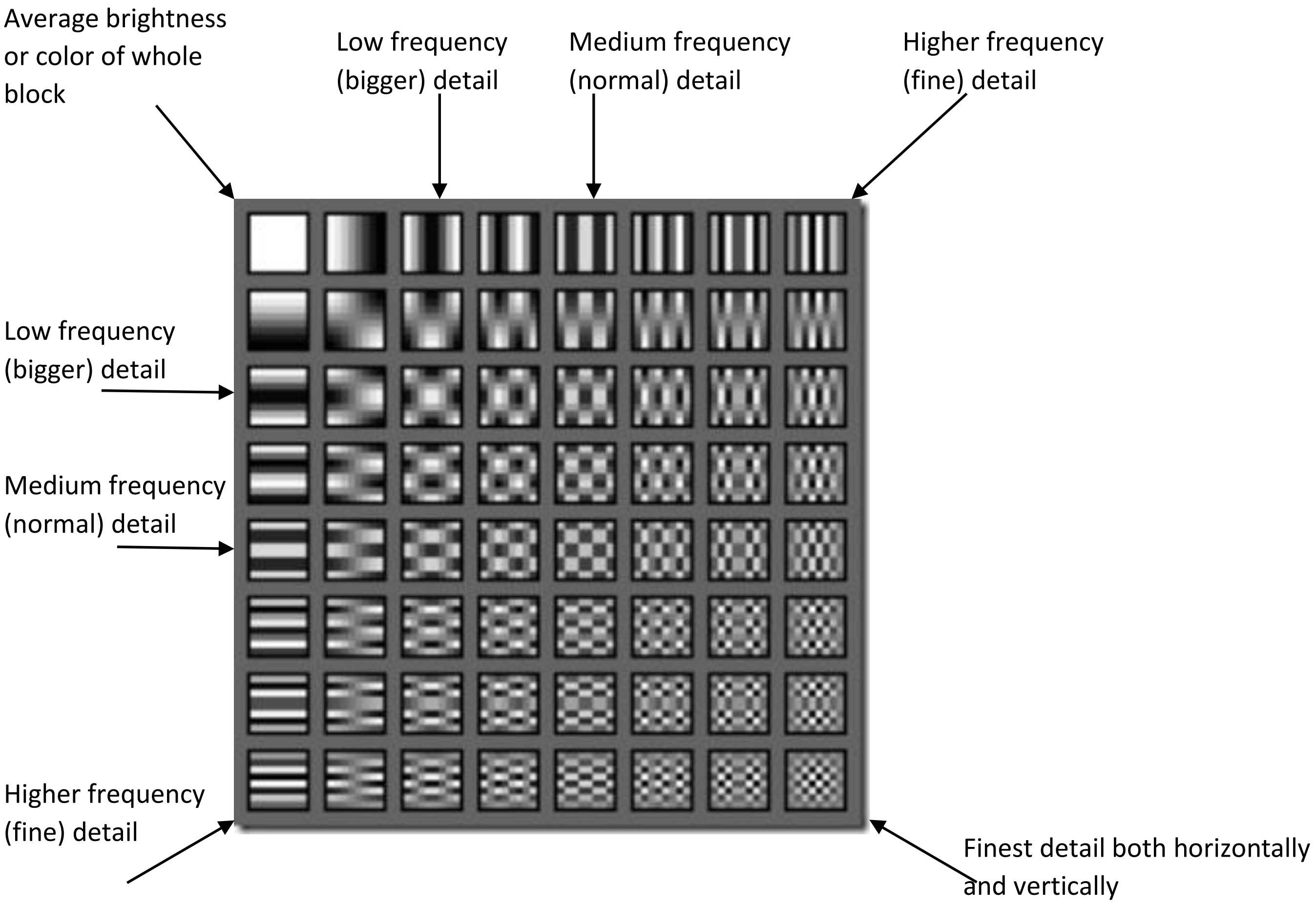

Average brightness or color of whole block
Low frequency (bigger) detail
Medium frequency (normal) detail
Higher frequency (fine) detail
Low frequency (bigger) detail
Medium frequency (normal) detail
Higher frequency (fine) detail
Finest detail both horizontally and vertically

Image Ref: http://www.digitude.net/blog/?p=8, http://forum.doom9.org/showthread.php?t=54147

Understanding the frequencies

1. *Average brightness*
   This is the average brightness and color of the whole image. All the other frequencies are relative to this average value. E.g. Average value depicts that the image is light red colored or it is medium bright.

2. *Low frequency*
   A big iron bar in the 8x8 block.

3. Medium Frequency
   3-4 striped lines in the block

4. High frequency
   Field, rains, hair have a lot of detail and can be categorized as high/very high frequencies.

A human visual system is barely able to notice the finer details in an image so such high frequencies can be compromised for compression.

Thus the Q matrix has higher values toward the bottom which be eventually zeroed for most of the images.

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Higher quantization values for finer details in an image

This clearly explains the selection of the Q matrix.

5. Varying the Quality factor

The degree of compression in JPEG can be varied, allowing a selectable trade-off between storage size and image quality. The user can decide the quality factor ranging from 1 to 100, where 1 gives the poorest image quality but highest compression and 100 provides the best quality but lowest compression.

For quality level other than 50, the quantization matrix at quality factor N is calculated as:

$$Q_N(i,j) = \begin{cases} \dfrac{100-N}{50} Q_{50}(i,j) & N > 50 \\ \dfrac{50}{N} Q_{50}(i,j) & N < 50 \end{cases}$$

Q_{10}	Q_{90}

80 55 50 80 120 200 255 305
60 60 70 95 130 290 300 275
70 65 80 120 200 285 345 280
70 85 110 145 255 435 400 310
90 110 185 280 340 545 515 385
120 175 275 320 405 520 565 460
245 320 390 435 515 605 600 505
360 460 475 490 560 500 515 495
	3 2 2 3 4 8 10 12
	2 2 2 3 5 11 12 11
	2 2 3 4 8 11 13 11
	2 3 4 5 10 17 16 12
	3 4 7 11 13 21 20 15
	4 7 11 12 16 20 22 18
	9 12 15 17 20 24 24 20
	14 18 19 19 22 20 20 19

Comparison of the DCT quantized matrix for lena.raw image with different quality factors for the first 8x8 block

Without quantization	**Quantization with QF = 50**

-8 -18 51 12 18 -67 -55 -27
55 -31 51 12 0 28 12 1
-32 78 -25 24 14 -7 12 5
20 2 -18 33 -19 15 -10 4
-4 -9 17 -29 7 -4 1 -6
-15 12 -24 15 -12 -4 0 0
5 -11 8 -13 5 -2 -2 -1
-2 2 0 2 -1 0 0 1
	-1 -2 5 1 1 -2 -1 0
	5 -3 4 1 0 0 0 0
	-2 6 -2 1 0 0 0 0
	1 0 -1 1 0 0 0 0
	0 0 0 -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

Quantization with QF = 10	**Quantization with QF = 90**

0 0 1 0 0 0 0 0
1 -1 1 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
	-3 -8 26 4 4 -8 -5 -2
	23 -13 18 3 0 2 1 0
	-11 30 -8 5 2 -1 1 0
	7 1 -4 6 -2 1 -1 0
	-1 -2 2 -3 1 0 0 0
	-3 2 -2 1 -1 0 0 0
	1 -1 1 -1 0 0 0 0
	0 0 0 0 0 0 0 0

The Q matrix values are quite high for Q_{10} while values for Q_{90} are quite low. Thus in the process of quantization most of the values quantized with QF = 10 are zeroed.

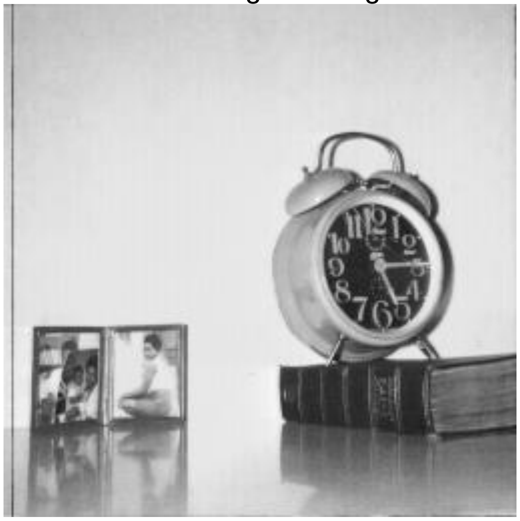
Thus we get a long stream of 0‘s for QF = 10 as compared to QF = 90. Thus lower QF matrices are best suited for entropy encoding given a very high compression at the cost of some degradation in quality.

3.4. JPEG Compression Quality Factor

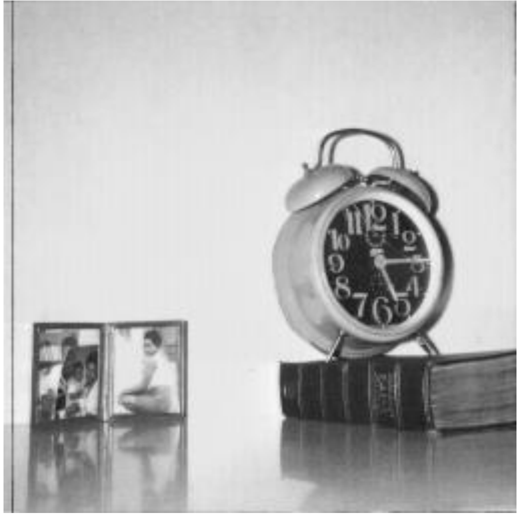
Public domain (http://www.ijg.org/) JPEG software jpeg_6b.zip is used to compress the images using JPEG standard.

In this section, we vary the quality factor and observe the compressed file sizes, PSNR and subjective quality of images.

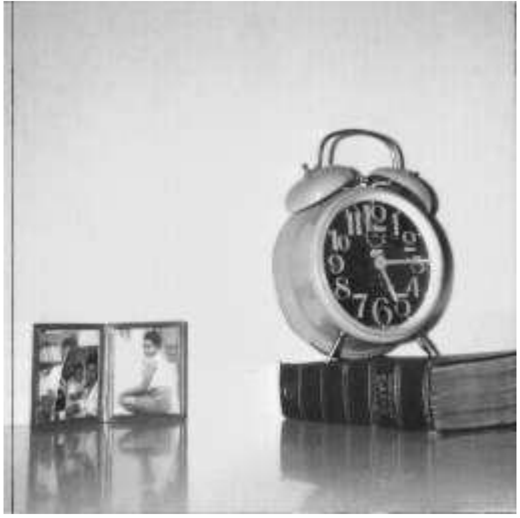
Original Image



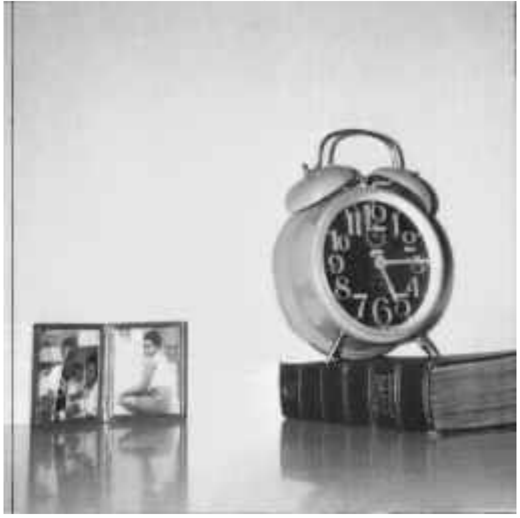
QF = 100



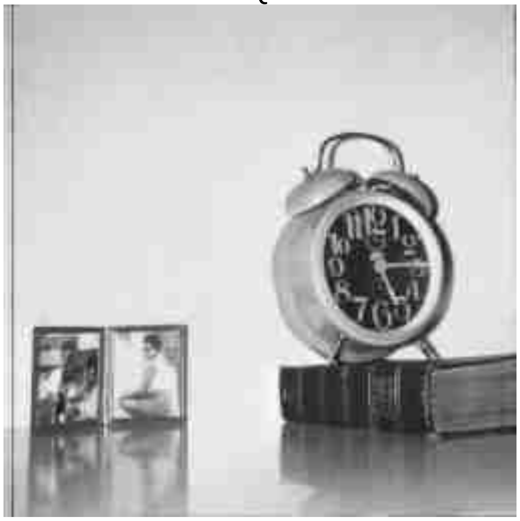
QF = 60



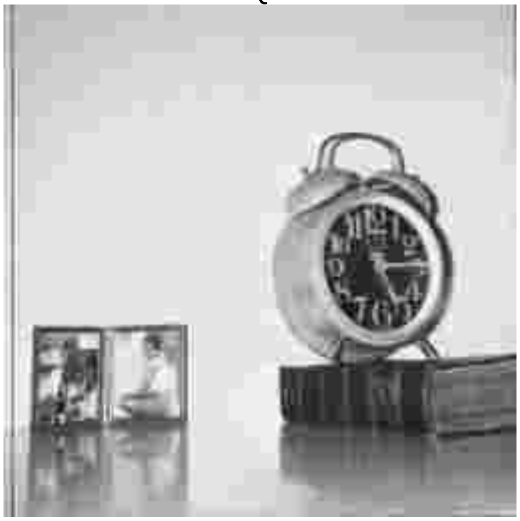
QF = 40

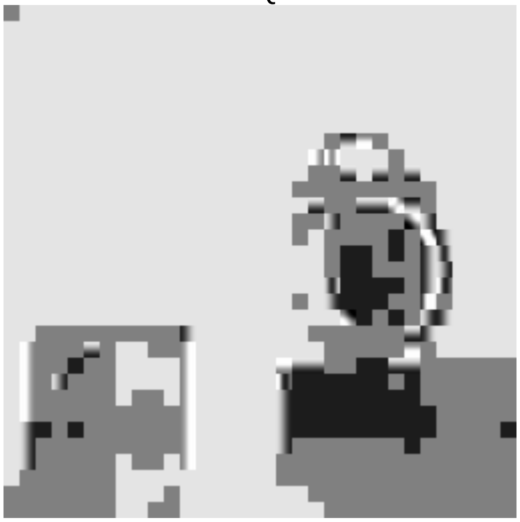


QF = 20



QF = 10





QF = 1

From the subjective quality of the images, we see that as the QF increases the quality of the image gets worse.

File name: *clock.bmp*, File size: *66,614* bytes			
Quality Factor	**Compressed File Size (bytes)**	**Compression Ratio = Original_file_size/ Compressed_file_size**	**PSNR**
100	37487	1.78:1	58.18
60	6795	9.80 :1	35.52
40	5440	12.25 :1	33.74
20	4045	16.47 :1	31.25
10	3152	21.13 :1	28.45
1	1847	36.07 :1	18.48

3.4.1 Discussion

- As shown in section 3.3 more the quality factor, better is the subjective image quality and lower is the compression ratio.
- The only lossy part in the JPEG Compression standard is the quantization. Higher values in the Q matrix will zero out many coefficients in the DCT matrix resulting in a higher compression.
- Quantization matrix is designed such that values are high for finer details that are less distinguishable to the human eye and are discarded. However with decrease in the quality factor, many of medium and low frequency components are also discarded resulting in loss of information which the human eye can make out.
- Low QF image appears as pixelated blocks for which the subjective quality of the image is quite bad.

3.5. Post-Processing of JPEG Encoded Images

From discussion in section 3.4.1, high compression comes at the cost of degraded quality of image. However, if we can improve the quality of the degraded images we can get good quality images with high compression,

Low QF image has pixelated blocks of data, so we can use a smoothing algorithm depending on the block image. One such algorithm used to do this task is:

Sung Deuk Kim ET all, “A Deblocking Filter with Two Separate Modes in Block-Based Video Coding”. IEEE transactions on circuits and systems for video technology, vol. 9, no. 1, February 1999.

3.5.1 Explanation of the algorithm

The algorithm has two separate filtering modes based on the local characteristics of the image.

- Smooth Region mode
  This mode is used when there is not much change in the pixel/brightness values in the block. In this mode, we gradually increase/decrease the intensity values so that there is a smooth transitioning and pixelated blocks are smoothened.
  E.g. if the stream of pixel values are as follows:

120	120	120	120	140	140	140	140	140	140

  This stream is categorized in smooth region mode and the smoothing operation will give the following output. [NOTE: This is not the exact output)

120	122	124	127	130	134	138	140	140	140

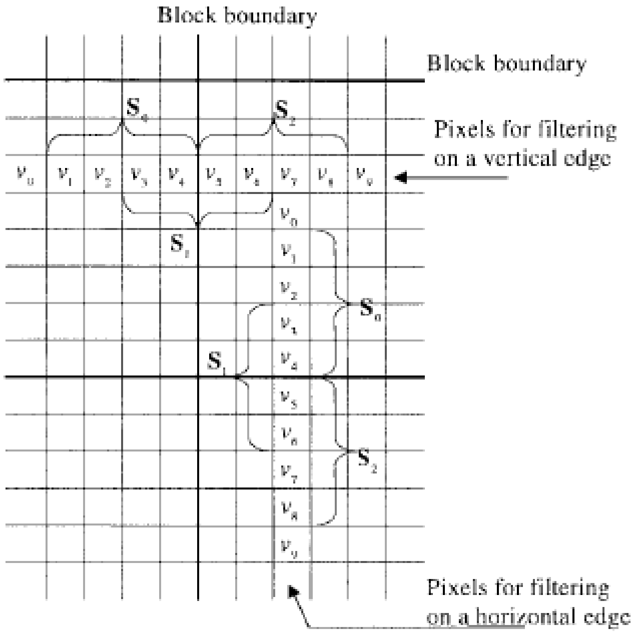
- Complex Region mode
  This mode is used when are there many variations in the stream processed. This stream corresponds to the finer details in the image. So directly smoothing the image will degrade the quality further. So we make use of DCT coefficient to check for finer details in the image. In this mode, only the border values are changed by a margin amount d, leaving the rest of the values unchanged.

  E.g. if the stream of pixel values are as follows:

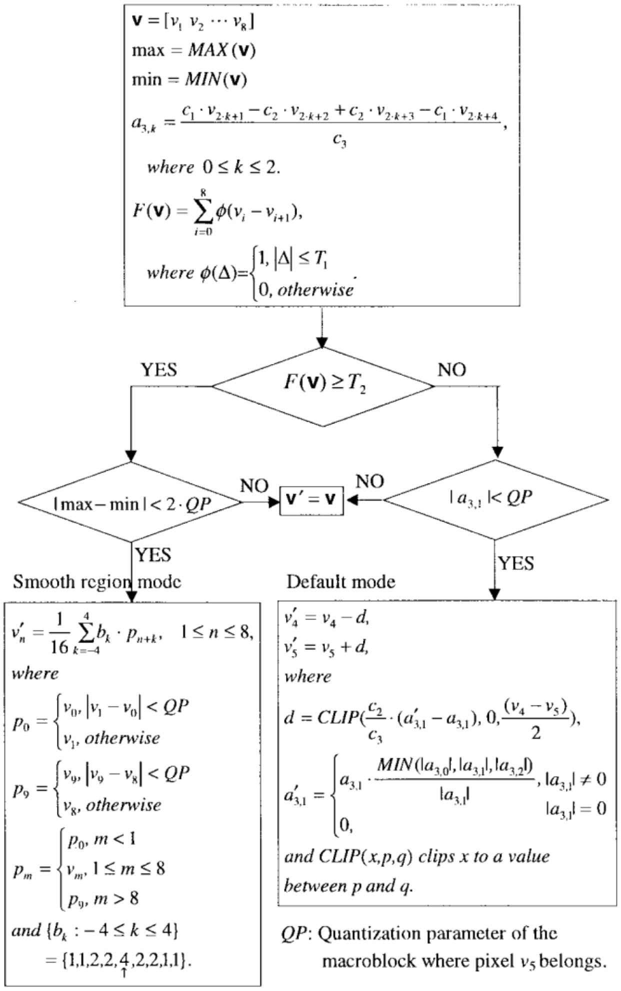
73	88	72	99	134	144	143	169	189	104

  Only the values in red will be changed by the algorithm keeping the finer details in the image intact.

We consider the stream of pixels as follows:



Block diagram of the whole process:

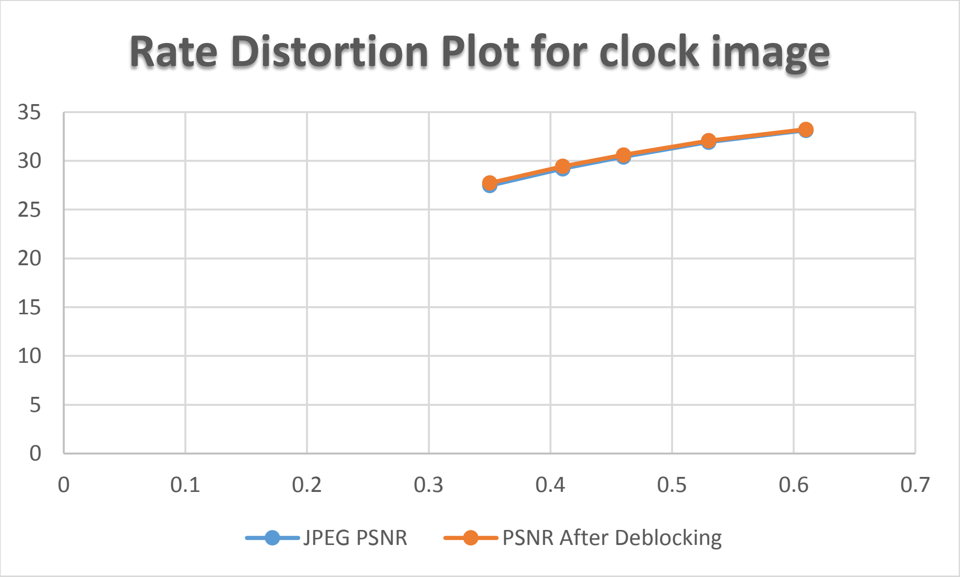


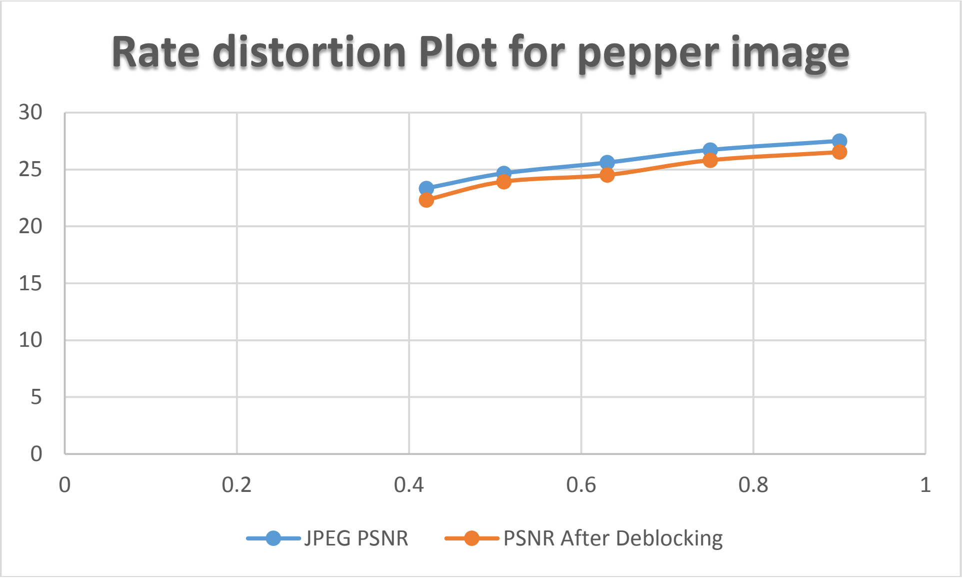
$\mathbf{v} = [v_1\ v_2 \cdots v_8]$
$\max = MAX(\mathbf{v})$
$\min = MIN(\mathbf{v})$
$a_{3,k} = \frac{c_1 \cdot v_{2\cdot k+1} - c_2 \cdot v_{2\cdot k+2} + c_2 \cdot v_{2\cdot k+3} - c_1 \cdot v_{2\cdot k+4}}{c_3},$
$where\ 0 \le k \le 2.$
$F(\mathbf{v}) = \sum_{i=0}^{8} \phi(v_i - v_{i+1}),$
$where\ \phi(\Delta) = \begin{cases} 1, & |\Delta| \le T_1 \\ 0, & otherwise \end{cases}.$
YES
$F(\mathbf{v}) \ge T_2$
NO
$|\max - \min| < 2 \cdot QP$
NO
$\mathbf{v}' = \mathbf{v}$
NO
$|a_{3,1}| < QP$
YES
YES
Smooth region mode
Default mode
$v'_n = \frac{1}{16} \sum_{k=-4}^{4} b_k \cdot p_{n+k}, \quad 1 \le n \le 8,$
$where$
$p_0 = \begin{cases} v_0, & |v_1 - v_0| < QP \\ v_1, & otherwise \end{cases}$
$p_9 = \begin{cases} v_9, & |v_9 - v_8| < QP \\ v_8, & otherwise \end{cases}$
$p_m = \begin{cases} p_0, & m < 1 \\ v_m, & 1 \le m \le 8 \\ p_9, & m > 8 \end{cases}$
$and\ \{b_k : -4 \le k \le 4\}$
$= \{1,1,2,2,\underset{\uparrow}{4},2,2,1,1\}.$
$v'_4 = v_4 - d,$
$v'_5 = v_5 + d,$
$where$
$d = CLIP(\frac{c_2}{c_3} \cdot (a'_{3,1} - a_{3,1}), 0, \frac{(v_4 - v_5)}{2}),$
$a'_{3,1} = \begin{cases} a_{3,1} \cdot \frac{MIN(|a_{3,0}|, |a_{3,1}|, |a_{3,2}|)}{|a_{3,1}|}, & |a_{3,1}| \ne 0 \\ 0, & |a_{3,1}| = 0 \end{cases}$
and $CLIP(x,p,q)$ clips x to a value between p and q.
QP: Quantization parameter of the macroblock where pixel v_5 belongs.

For more details on the algorithm, refer the paper.

3.5.2 Rate Distortion curves

Image	Bits/pixel	PSNR of JPEF file	PSNR of JPEG file after deblocking
Clock1	0.35	27.46	27.73
Clock2	0.41	29.19	29.43
Clock3	0.46	30.41	30.60
Clock4	0.53	31.92	32.07
Clock5	0.61	33.13	33.23
Pepper1	0.42	23.34	22.32
Pepper2	0.51	24.66	23.93
Pepper3	0.63	25.60	24.51
Pepper4	0.75	26.71	25.81
Pepper5	0.9	27.50	26.52



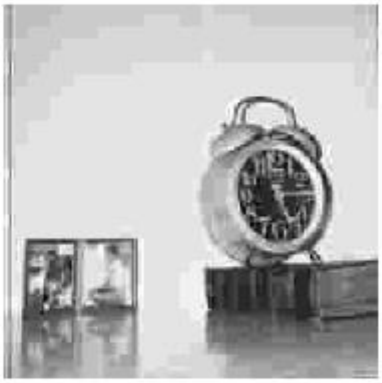


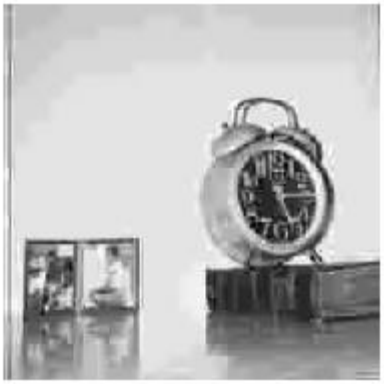
3.5.3 Comparing the subjective quality of the images

For black and white images, the procedure for Deblocking is the same as described in section 3.5.1.

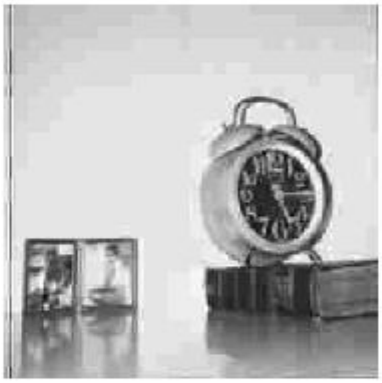
However for colored images, following steps have been taken:

1. Convert the RGB image format to YUV.
2. Apply the Deblocking algorithm only on the Y component.
3. Modify the Y component, merge it with the U and V components.
4. Convert YUV back to RGB format.

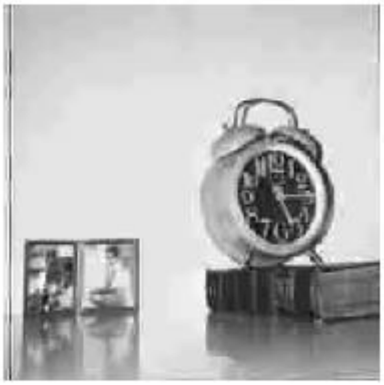

clock1 JPEG Image


clock1 JPEG Image after deblocking

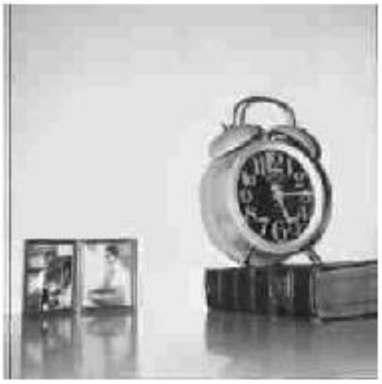
clock2 JPEG Image



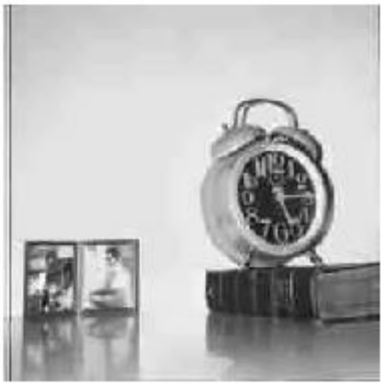
clock2 JPEG Image after deblocking



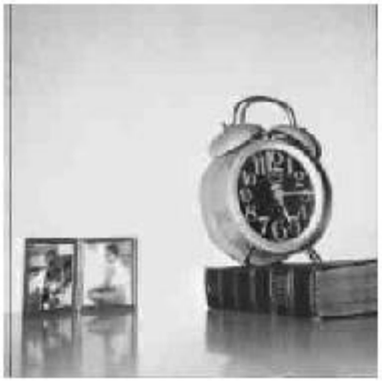
clock3 JPEG Image



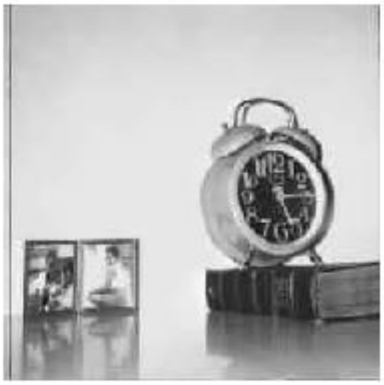
clock3 JPEG Image after deblocking



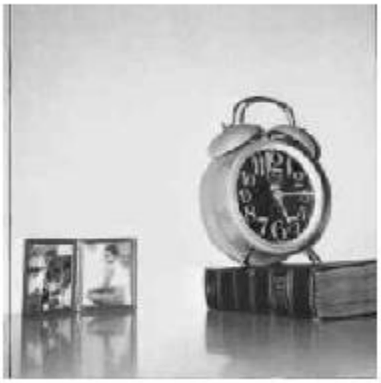
clock4 JPEG Image



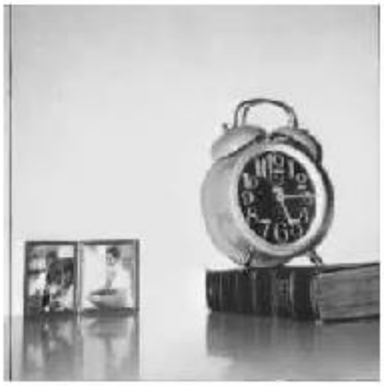
clock4 JPEG Image after deblocking



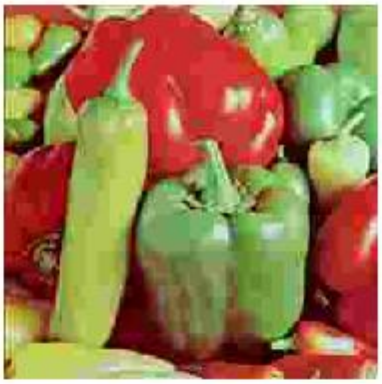
clock5 JPEG Image



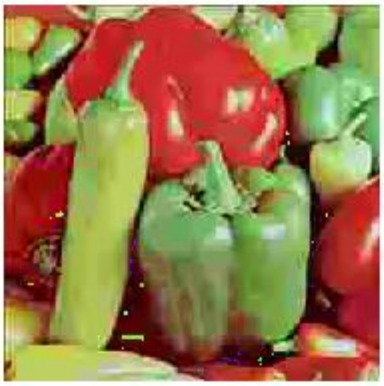
clock5 JPEG Image after deblocking



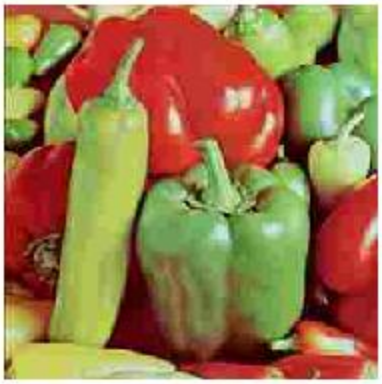
pepper1 JPEG Image



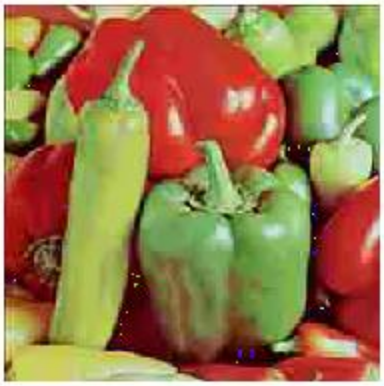
pepper1 JPEG Image after deblocking



pepper2 JPEG Image

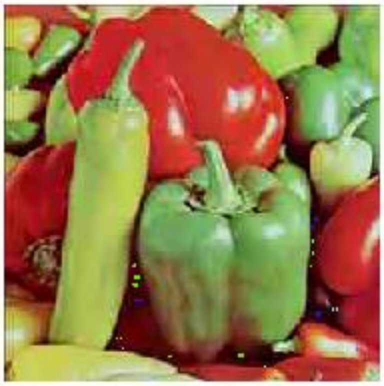


pepper2 JPEG Image after deblocking



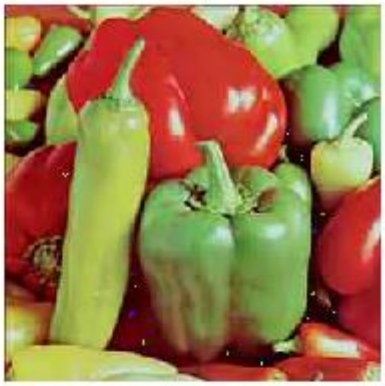
pepper3 JPEG Image

pepper3 JPEG Image after deblocking



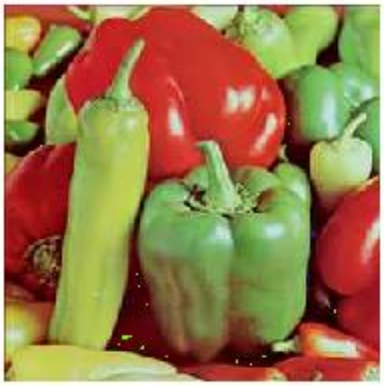
pepper4 JPEG Image

pepper4 JPEG Image after deblocking



pepper5 JPEG Image

pepper5 JPEG Image after deblocking



3.5.3.1 Discussion

- The images after Deblocking have a better subjective quality than the JPEG images. The pixelated blocks in the original JPEG images have been smoothened to a good extent
- For colored images, due to the application of the Deblocking algorithm only to Y component there are some colored spots appearing in the image (pepper images). However the overall quality of image is better than JPEG image.

- There is not much difference between the PSNR of the JPEG and deblocked JPEG file. In some cases, the PSNR of the deblocked JPEG file is less than JPEG file even though the subjective quality of the deblocked file is better. This is because the PSNR is not the best measure to check for the image quality. PSNR is based on the MSE which just compares the original pixel values with the quantized values. Even though the MSE for the deblocked images are high for some instances due to high difference between original and deblocked image, the smoothening factor in the deblocked image makes the quality of image as perceived by the human eyes quite better.