EEE 330 Assignment 4

Shuchang Fan

1201374

2017.5.9

Answer:

The difference of this DCT compression engine between JPEG is listed as the following table, the 2D-DCT, **Quantization and Entropy coding** process are shared by both of the two methods, and the main difference is that JPEG is using a **zig-zag** reading to achieve larger runs of zeros. The 'zig-zag' reading reads the data stored in the matrix in the shape of letter 'Z'. While for DCT, it reads the data column by column. Hence, it is possible to achieve large runs of zeros by zig-zag reading. So, fewer bits are used to encode it, indicating higher compression ratio.

	DCT engine	JPEG
2D-DCT	YES	YES
Quantization	YES	YES
Zig-Zag	**NO**	**YES**
Entropy Coding	YES	YES

Table 1 Compression progres

Figure 1 Zig-Zag

The Q_{mat} matrix and mathematical explanation is shown as the following figure

$$\mathrm{Qmat} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad (1)$$

The process can be described as

$$Q_{ij} = round\left({b_{ij}} / {S * q_{ij}} \right)$$

While S is determined by,

$$S = \frac{100 - QP}{50}, QP > 50$$

$$S = \frac{50}{QP}, QP \le 50$$

The QP is a representation of the quantization level, the higher the QP, the heavier of the compression will be.

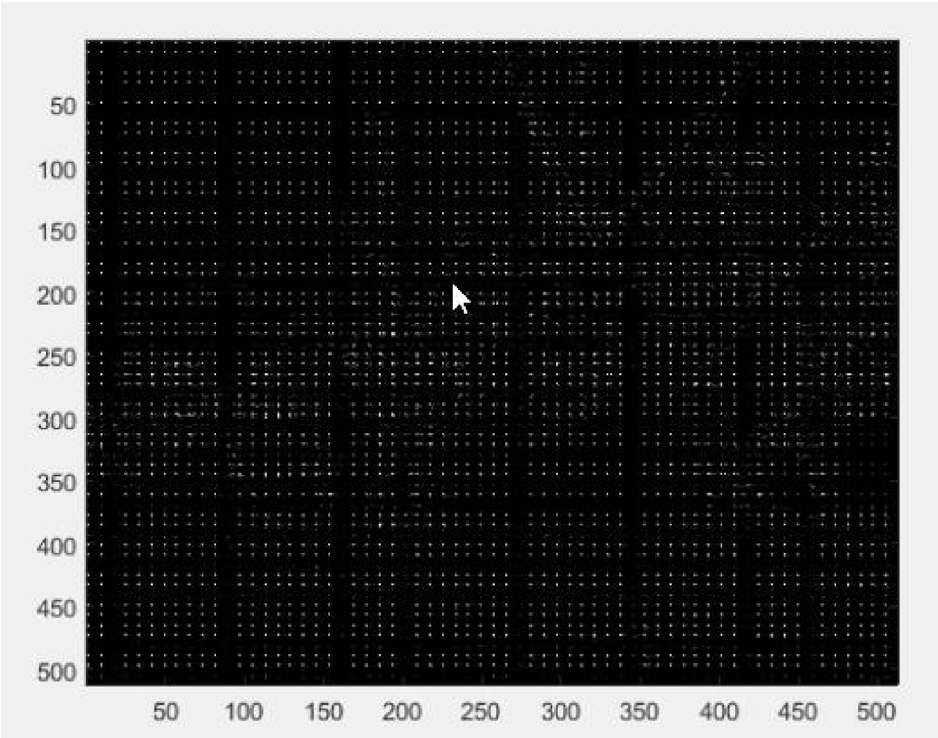


Figure 2 DCT transformed image

As for QP=50, in order to obtain the bit rate, the value obtained should multiply by 8 to transfer into bits and then divided by the total size, the calculated rate is 1.2024bits/pixel.

```
Arith07: Sequence 1 of 262144 symbols encoded using 317549 bits.

rate =

    1.2024

Elapsed time is 22.203125 seconds.
```

```
function [rate] = compress_im(im, Qmat, QP, file_name)
[r, c]=size(im);

if QP>50
S=(100-QP)/50;
else
S=50/QP;
end

for i=1:8:r-7 ;
for j=1:8:c-7 ;
A=im(i:i+7, j:j+7);
im_dct=dct2(A);
im_dct_2(i:i+7, j:j+7)=im_dct;
im_quantization(i:i+7, j:j+7)=round(im_dct./(S*Qmat));
end
end

figure;

imshow(uint8(im_dct_2));
colormap(gray(255));
```

```
clear all;
close all;
clc;

tic
im=imread('boat.512.tiff');
Qmat=[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];
 QP=50;
 file_name ='file_name.m';
[rate] = compress_im( im, Qmat, QP, file_name );
toc
```

2. Image decompression **(10 marks)**

   Write a Matlab function which decompresses the file `file_name` and returns the corresponding image `imo`. The function should have the following declaration:

   ```
   function [imo] = decompress_im(Qmat, QP, file_name,..
   ```

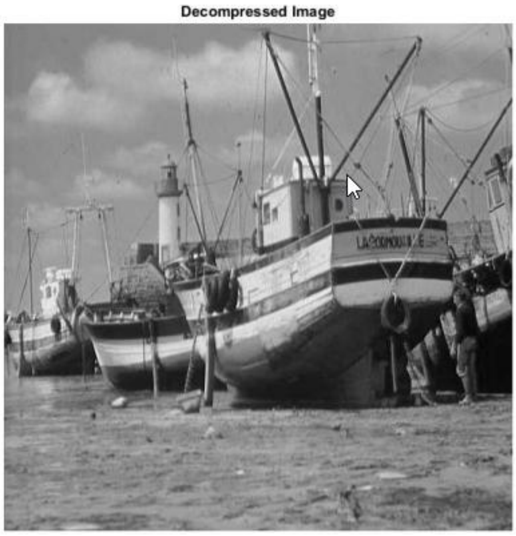
   This function should invert all steps in task 1 to produce the decompressed image `imo`.

Answer:

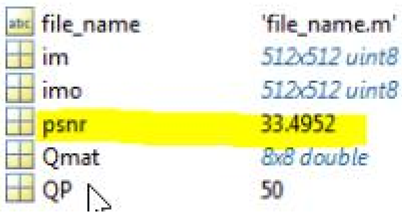
The following fucntion will operate the exactly opposite function compared to the previous compression one. Firstly, the enctropy decoder was used to decode the matrix which will be used to construct the image. Secondly, reshaping the matrix to a 512*512 sized. The process can be described using the following equation.

$$DQij=bij*S*qij$$

The reconstruction result is like the following figures,



In order to evaluate the decompression quality, the PSNR was introduced. Where the PSNR value is 33.4952.



file_name	'file_name.m'
im	512x512 uint8
imo	512x512 uint8
psnr	33.4952
Qmat	8x8 double
QP	50

```
function [ imo ] = decompress_im( Qmat, QP, file_name )
r=512;
c=512;
Fp=fopen(file_name, 'rb');
E=fread(Fp);
fclose(Fp);
e_dec=entropy_dec(E);
A=reshape(e_dec, [r c]);

if QP>50
S=(100-QP)/50;
else
S=50/QP;
end

for i=1:8:r-1
for j=1:8:c-1
im_dec=A(i:i+7, j:j+7);
im_dec_Q=(S*Qmat).*im_dec;
imo(i:i+7, j:j+7)=idct2(im_dec_Q);
end
end

imo=uint8(imo);
end
```

```
clear all;
close all;
clc;
tic
im=imread('boat.512.tiff');
Qmat=[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];

QP=50;
file_name =sprintf('file_name.m');
imo = decompress_im( Qmat, QP, file_name );
toc

figure;
imshow(im);
title('Original Image');
figure;
imshow(imo);
title('Decompressed Image');
```

3. Rate-Distortion (RD) Performance evaluation (30 marks)

Calculate the PSNR of the reconstructed image when the original image is Lenna512.bmp. Please use several QP values (say for example QP=[1:14:99]), and evaluate the corresponding bitrate. Plot the PSNR versus bit rate curve for the image Lenna512.bmp; it is worth noticing that this curve is wrongly and commonly called RD curve in many resources. **(10 marks)**

Compare the Rate-PSNR performance of your codec with that of Jpeg and JPEG2000. For the latter two codecs you could use the Matlab command imwrite with proper parameters. Instead of using the quantization matrix Qmat in your codec use a uniform quantization matrix, for example $16 \times \mathrm{ones}(8)$. Evaluate the Rate-PSNR curve of this new configuration and compare it with the one obtained with Qmat. In comparing this, justify which approach provides a better output. What are the benefit(s) of using a uniform quantization matrix over the one given in (1)? **(10 marks)**

Propose some approaches to enhance the RD efficiency of your codec, and justify your proposal. **(10 marks)**

Answer:

3.1 Evaluate the PSNR performance (DCT engine)

Firstly, different values of QP were used, corresponding after the compression process, the different values of bit rate were produced. Similarly, a comparison was made between different pairs of reconstructed image and original images. Thus we could obtain a relationship between the bit rate and PSNR. With the increase of QP, the image becomes clearer. This is consistent with the increasing of PSNR recorded in Table 1 below.

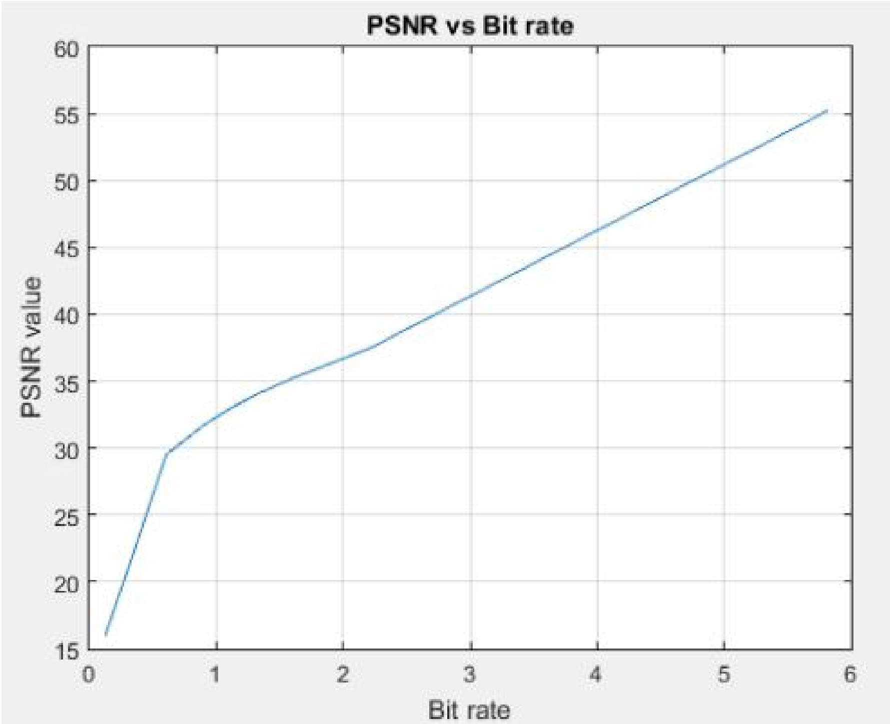


Figure 3 bit rate vs PSNR

OP	1	15	29	43	57	71	85	99
PSNR	15.9509	29.5281	31.7205	33.0053	33.9819	25.2228	37.4737	55.2577
Bit rate	0.1264	0.6094	0,9017	1.1179	1.3110	1.6014	2.2130	5.8260

Table 2 bit rate vs PSNR-DCT engine

3.2 Evaluate the PSNR performance (JPEG)

QP represents the quality factor and ranges from 1 to 99. When QP=1, the image is badly distorted and PSNR is only around 23. With the increase of QP, the corresponding PSNR increased, indicating the higher quality of the compressed image.

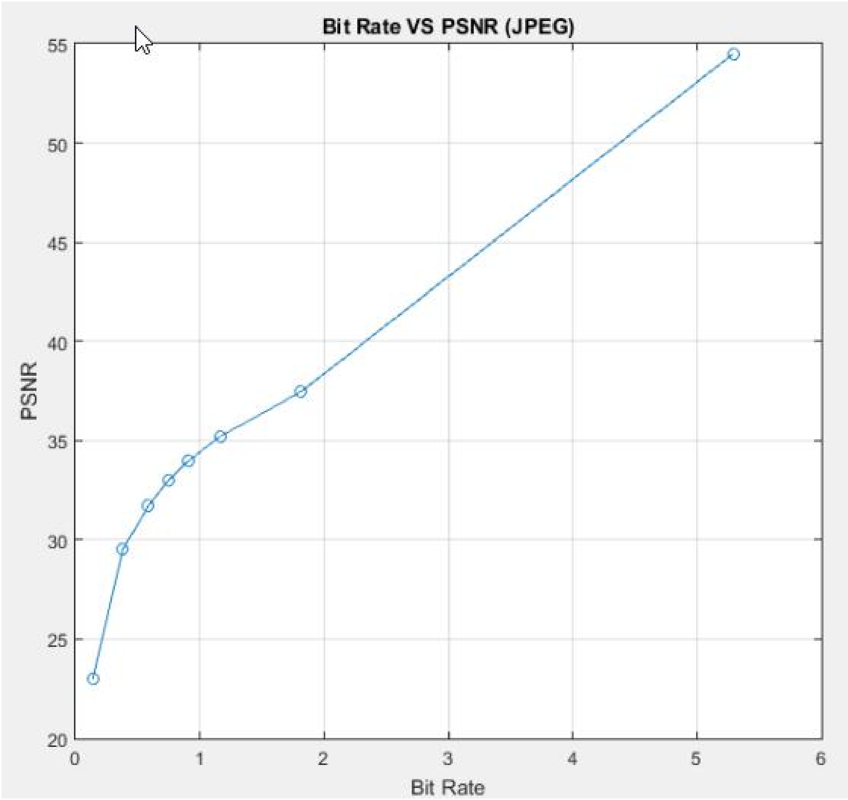


Figure 4 bit rate vs PSNR (JPEG)

OP	1	15	29	43	57	71	85	99
PSNR	22.9877	29.5252	31.7169	33.0024	33.9854	35.2261	37.4607	54.4986
Bit rate	0.1379	0.3824	0.5858	0.7486	0.9080	1.1685	1.8142	5.2955

Table 3 bit rate vs PSNR-JPEG

3.2 Evaluate the PSNR performance (JPEG2000)

Instead of using DCT compression method, the jpg2000 introduced the wavelet compression method. Hence, the compressed image compressed more efficiently. In addition, there is no correlation of each 8*8 blocks, they are independent with each other. Therefore, for large compression rate, there will be 'blocky' in the recovered image. For jpg2000, there is no blackness whatever the compression ratio is.

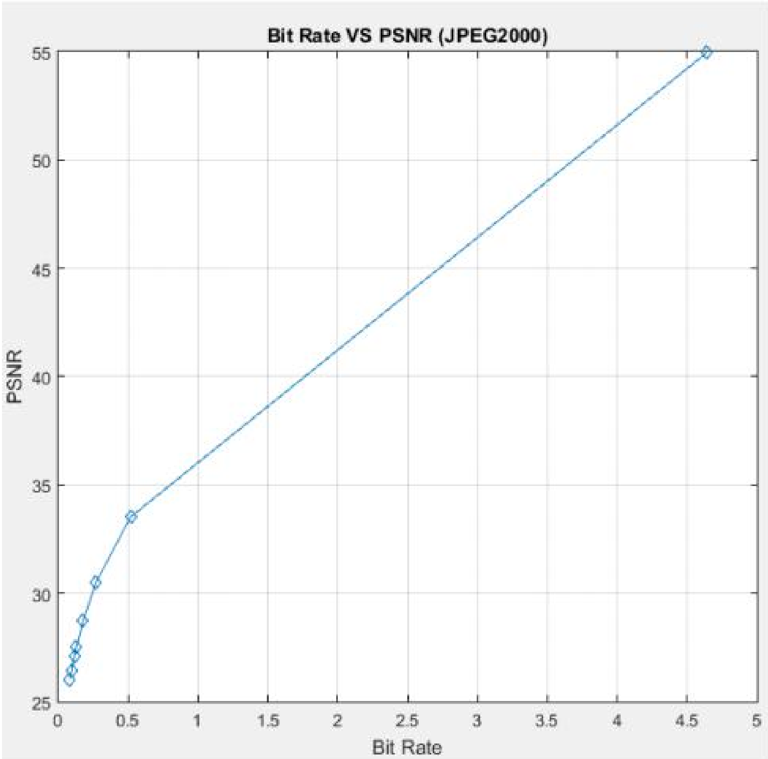


Figure 5 bit rate vs PSNR-JPEG2000

OP	1	15	29	43	57	71	85	99
PSNR	54.9432	33.5171	30.4745	28.7195	29.4917	27.0861	26.4206	25.9890
Bit rate	47.6378	0.5173	0.2688	0,1782	0,1301	0,1149	0,0952	0.0825

Table 4 bit rate vs PSNR-JPEG2000

3.3 Evaluate the PSNR performance (DCT, JPEG, JPEG2000)

Comparing DCT engine with JPEG and JPEG2000, JPEG2000 always has best PSNR for given bit rate as shown in Figure 6, it is totally above the other two approaches. In other words, given bit rates the PSNR of Jpeg2000 compression is always better than Jpeg compression and DCT.

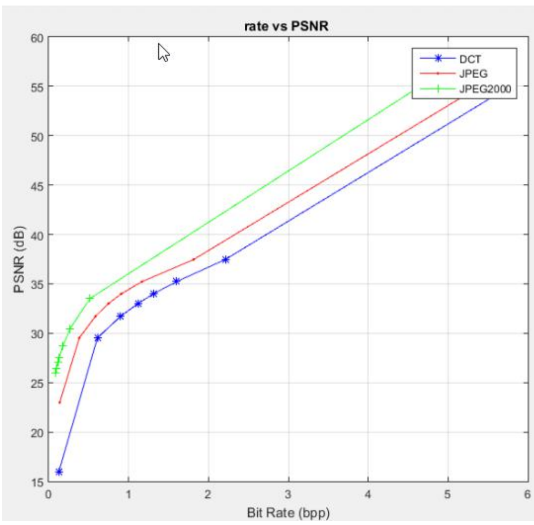


Figure 6 comparison of three approaches

3.4 Compare Qmat & uniform Quantization

It is surprising to see that the performance of uniform quantization surpass the given Qmat which is unexpected.

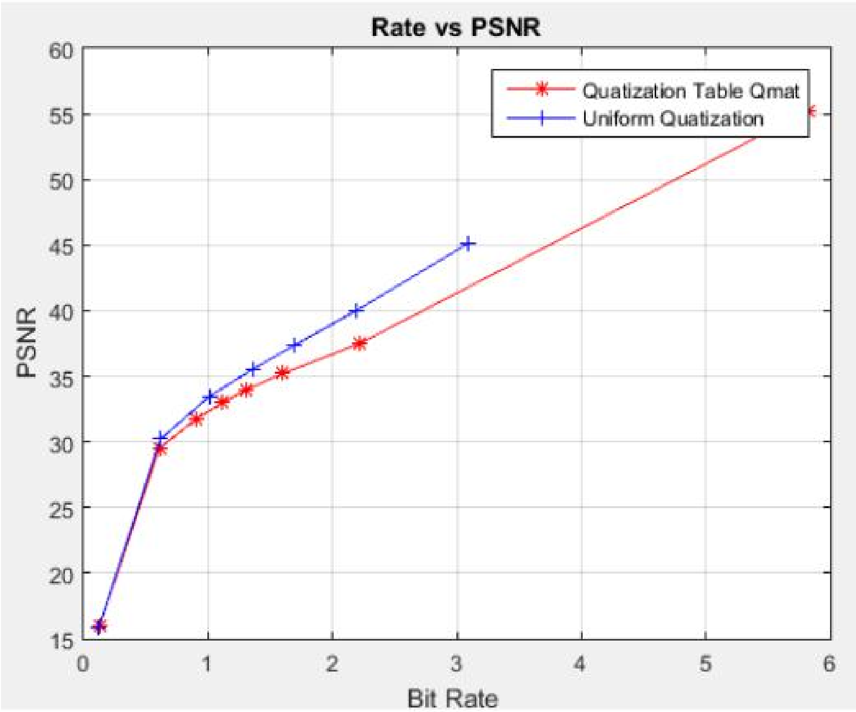


Figure 7 Comparison between two Q_{mat}

As shown in Figure 7 above, when bit rate is low (around 0.5bpp) the PSNR of DCT using quantization table Q_{mat} and uniform quantization are almost the same. As a while for bit rate lager than 0.5 bpp, given the same bit rate, the PSNR of DCT engine compressed image with uniform quantization is better than that with quantization table Q_{mat}. It can only approved that under some specific circumstances, the uniform quantization has a better performance.

This phenomenon happened for reasons:

1. The input image 'boat.512.tiff' is actually a quite smooth picture, with no rapid changing edges--not much high frequency components.
2. 2. Zig-zag reordering is absent.

The Q_{mat} applied different gap size corresponding to different frequencies, if the testing samples are large enough, the performance of the Q_{mat} will be justified.

3.5 proposal to enhance the RD efficiency

Image information consist of DC component an AC component, during our design, there is no difference when considering coding DC component or AC component.

As for some regions, the DC component outnumbered the AC component which is the most common case. Thus we could use different coding scheme to separate these two components.

Although DC components are large, but the changes between each block are not that significant. Thus we can design a scheme with fewer bits to encode the differences to enhance the RD efficiency.

```
clear all;close all;clc;
tic
k=1;
i=1;
j=1;
tic

%%
im=imread('boat.512.tiff');
imo=[];
Qmat=[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];
file_name='boat.mat';
file_name_2='boat.512JPEG.jpg';
file_name_3='boat.512JPG2000.jpg';
%% DCT entropy decoder
for QP=1:14:99
[rate]=compress_im(im,Qmat,QP,file_name);
[imo] = decompress_im(Qmat, QP, file_name);
QP_all(k)=QP;
rate_all(k)=rate;
PSNR=psnr(imo,im);
PSNR_all(k)=PSNR;
k=k+1;
end
figure(1),
plot(rate_all, PSNR_all, 'marker', '*'); xlabel('Bit Rate');
ylabel('PSNR');title('Bit Rate VS PSNR');
grid on
```

```
%% JPEG
for QP=1:14:99
imwrite(im,file_name_2,'jpeg','Quality',QP);
imo_2=imread('boat.512JPEG.jpg');
[size_2] = fsize(file_name_2);
rate_2=(size_2*8)/(512^2);
size_all_2(i)=size_2;
rate_all_2(i)=rate_2;
PSNR_2=psnr(imo_2,im);
PSNR_all_2(i)=PSNR_2;
i=i+1;
end
figure(2),
plot(rate_all_2, PSNR_all_2, 'marker', 'o'); xlabel('Bit Rate');
ylabel('PSNR');title('Bit Rate VS PSNR (JPEG)');
grid on
%% JPEG 2000
for QP=1:14:99
imwrite(im,file_name_3,'jp2','CompressionRatio',QP);
imo_3=imread('boat.512JPG2000.jpg');
[size_3] = fsize(file_name_3);
rate_3=(size_3*8)/(512^2);
size_all_3(j)=size_3;
rate_all_3(j)=rate_3;
PSNR_3=psnr(imo_3,im);
PSNR_all_3(j)=PSNR_3;
j=j+1;
end
figure(3),
plot(rate_all_3, PSNR_all_3, 'marker', 'diamond'); xlabel('Bit
Rate'); ylabel('PSNR');title('Bit Rate VS PSNR (JPEG2000)');
grid on

toc
%%
figure(4),
plot(rate_all,PSNR_all,'b*-',rate_all_2,PSNR_all_2,'r.-
',rate_all_3,PSNR_all_3,'g+-');
legend('DCT','JPEG','JPEG2000')
title('rate vs PSNR ');
grid on;xlabel('Bit Rate (bpp)');yl
```

```
%% comparison of Qmat with uniform quantization matrix in terms of
rate vs psnr
figure,plot(rate_set,PSNR_Set_dct,'r*-
',rate_set_dct_UQ,PSNR_Set_dct_UQ,'b+-');
title('Rate vs PSNR');
grid on;xlabel('Bit Rate ');ylabel('PSNR');
legend('Quatization Table Qmat','Uniform Quatization');
```