

实验一 熟悉 JPEG 压缩过程

学号：201711123010 班级：信息安全 2017 级 姓名：湛雯馨

1. 实验目的

熟悉 Matlab/c/python 图像处理编程环境。

模拟并熟悉图像 JPEG 压缩过程。

2. 实验环境

本次实验所使用的图像是 512x512 的灰度图像（即 lena512.bmp）使用的工具及

版本为 MATLAB R2017B。

3. 实验内容

3.1 模拟数字图形 jpeg 压缩，绘制 PSNR~Q 曲线

1) 实验过程

1.调用 matlab 内带函数 imwrite 设置质量因子为 1-100，进行 jpeg 压缩，生成 jpeg 图像

2.读入 100 幅图像，并分别计算 PSNR

3.通过 matlab 函数 plot()将其图像画出

4. 实验代码：(包含注释)

```
1. close all;clear all;clc;
2. I=imread('lena.bmp');           %读入图片
3. x=zeros(1,100)                   %初始化一个 100 个元素的一维矩阵
4. for i=1:100
    imwrite(I,[sprintf('%03d',i),'lena.jpg'],'quality',i);    x(i)=i
end                                %不同质量因子生成不同的图片
5. I_double=double(I)               %强制类型转换
6. PSNR=zeros(1,100)                %初始化 PSNR 数组
7. for i=1:100
    J1=imread([sprintf('%03d',i),'lena.jpg'])
    J1_double=double(J1)
    D=J1_double-I_double%PSNR 计算公式
```


















```

MSE = sum(D(:).*D(:)) / numel(I); %numel 计算数组中的元素个数
PSNR(i) = 10*log10(255^2 / MSE);
end
8. plot(x,PSNR) %画出 PSNR-Q 图像

```

2) 实验结果

1. 生成了质量因子不同的 jpeg 压缩图像

| 名称 | 修改日期 | 类型 | 大小 |
|---|----------------|----------------|-------|
|  001lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 5 KB |
|  002lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 5 KB |
|  003lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 5 KB |
|  004lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 6 KB |
|  005lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 6 KB |
|  006lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 7 KB |
|  007lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 7 KB |
|  008lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 7 KB |
|  009lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 8 KB |
|  010lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 8 KB |
|  011lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 9 KB |
|  012lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 9 KB |
|  013lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 10 KB |
|  014lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 10 KB |
|  015lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 10 KB |
|  016lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 11 KB |
|  017lena.jpg | 2019/9/26 8:11 | WPS看图 JPG 图... | 11 KB |

2. 绘制出了 PSNR-Q 的曲线图

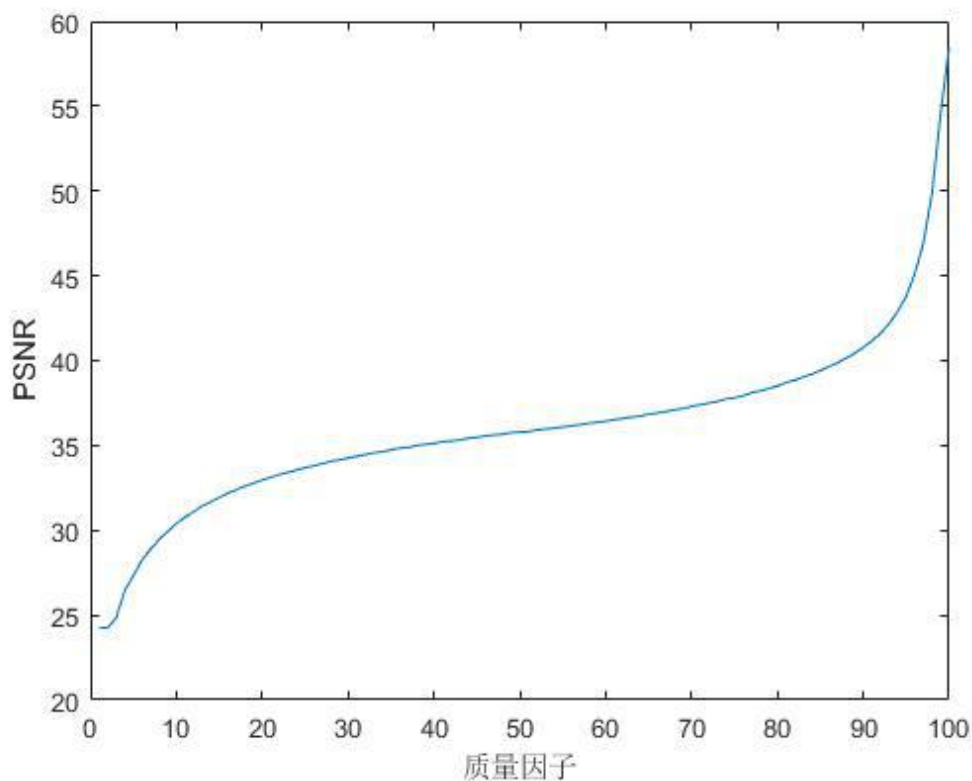


图 1 不同质量因子对应的 PSNR 图

3.2 显示压缩前后的灰度直方图，观察并分析所存在的差异

1) 实验过程

1. 本次实验选取了质量因子为 50, 70, 75, 80 的 jpeg 图像作为样本，调用 matlab 中函数 `imhist` 生成灰度直方图。

2. 同样通过调用 `imhist` 函数对原图生成灰度直方图。以下链接是 `imhist` 函数详解：

https://ww2.mathworks.cn/help/images/ref/imhist.html?s_tid=srchtitle

3. 实验代码：

```
1. J1=imread('050lena.jpg')
2. imhist(J1);                                %[counts,binLocations] = imhist(I) 计算灰度
图像的直方图 I。该 imhist 函数返回中的直方图计数 counts 和中的箱位置 binLocations。直方
图中的 bin 数由图像类型决定。
3. figure;                                    % 画图
```

2) 实验结果

1. 选取的六幅灰度图样例

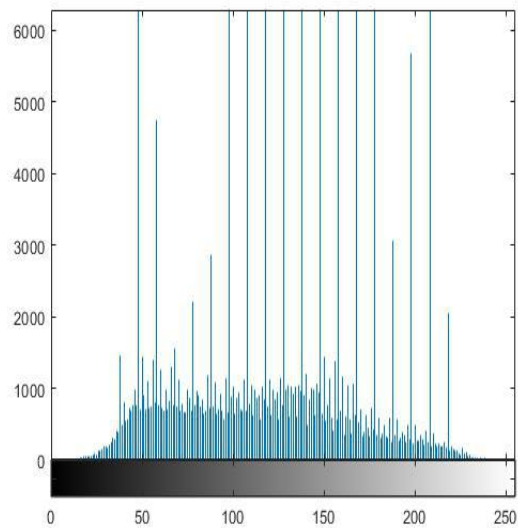


图 2 质量因子 10 的灰度直方图

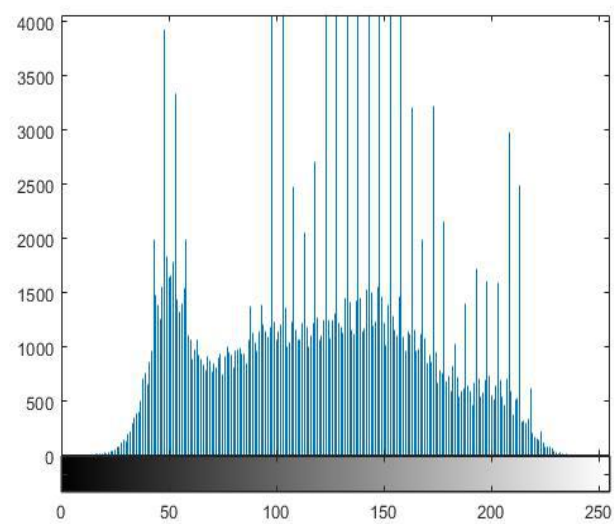


图 3 质量因子 20 的灰度直方图

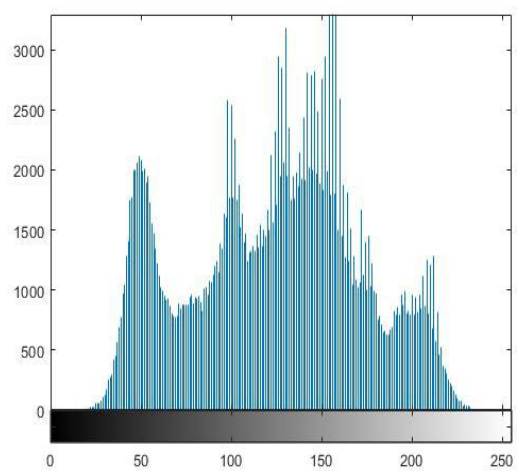


图 4 质量因子 50 的灰度直方图

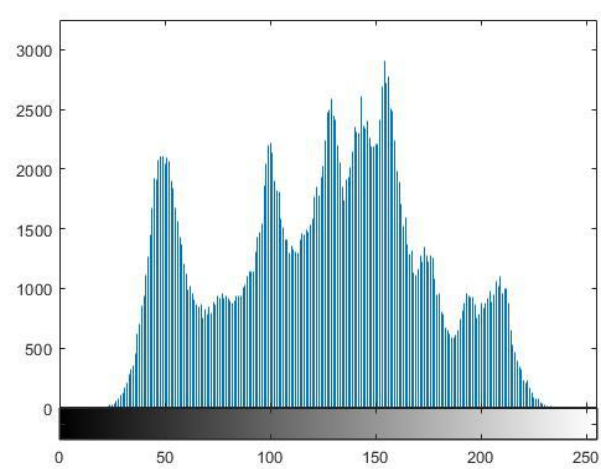


图 5 质量因子 70 的灰度直方图

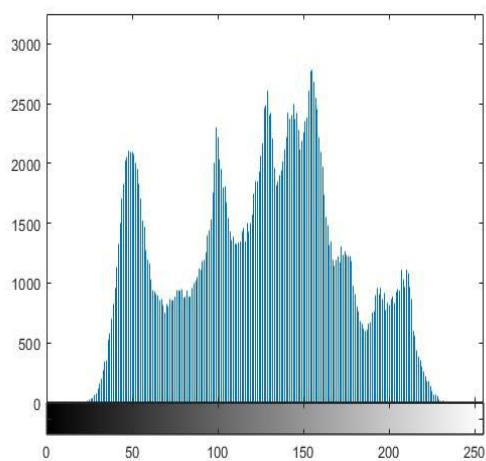


图 6 质量因子 75 的灰度直方图

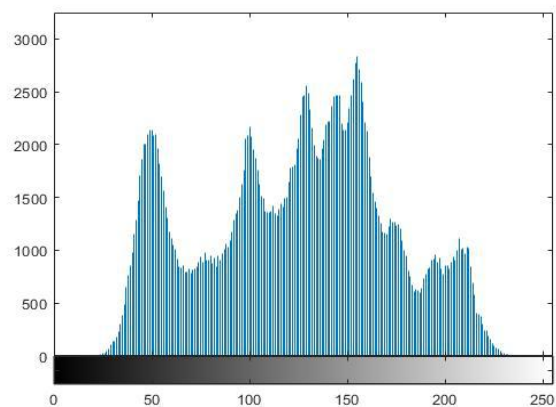


图 7 质量因子 80 的灰度直方图

2. 原图的灰度直方图

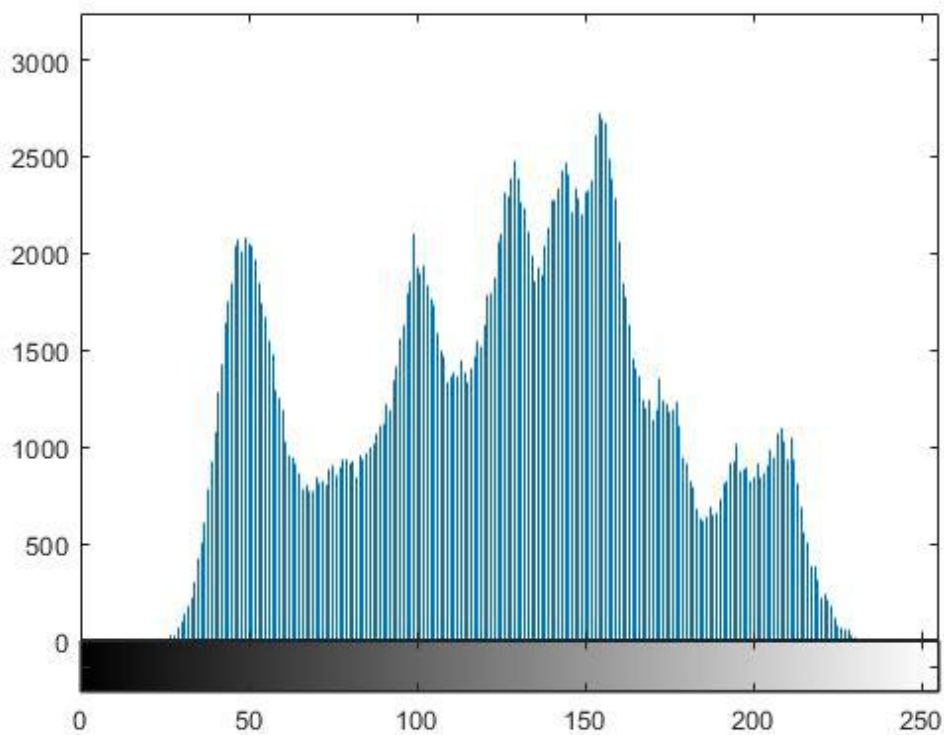


图 6 原图的灰度直方图

3. 观察并分析所存在的差异

由观察可知，质量因子靠近 70 左右的图像的灰度直方图与原图的灰度直方图差异不大，质量因子越小，灰度直方图差异越明显，压缩率也更大。质量因子较小时，灰度直方图中出

现大片像素的频率很小，而某些像素出现的频率很大的情况，质量因子越小，这种情况越明显。而质量因子越大，与原图的灰度直方图越接近。而观察生成的图像，质量因子越小，边界处越模糊，灰度级越少。

3.3 读取 JPEG 图像文件，自己编解码提取第 x 个宏块的量化后 dct 系数，模拟反量化和逆 DCT 变换，恢复并显示对应空域图形块，观察并分析 JPEG 压缩所引起的块效应。

1) 实验过程

1. 读取 JPEG 图像文件，使用 `jpeg_read` 可以同时读取量化表，dct 系数等等变量
2. 对图像进行分块，并选取第 10 个宏块
3. 对该宏块进行反量化
4. 逆 dct 变换
5. 恢复图像块
6. 实验代码

```
1. I=imread('lena.bmp')           %读取图像矩阵，此处选取质量因子为 50 的 jpeg 图像
2. H=512;L=512;i=0;
for height=1:H/64%进行 64*64 分块
    for length=1:L/64
        i=i+1;
        I_block_eight(:, :, i)=I(((height-1)*64+1):((height-1)*64+64),((length-1)*64+1):((length-1)*64+64));%行-->--,列-->--
    end
end
3. origin=I_block_eight(:, :, 10)    %选取原图像的第十个宏块
4. imshow(origin) figure;           %把原图像的宏块显示出来
5. COVER='050lena.jpg'; jobj = jpeg_read(COVER); % JPEG image structure
6. DCT = jobj.coef_arrays{1,1};      % DCT plane
    Y_Table=jobj.quant_tables{1,1}; %量化表
7. i=0; for height=1:H/64            %进行 64*64 分块
    for length=1:L/64
        i=i+1;
        block_eight(:, :, i)=DCT(((height-1)*64+1):((height-1)*64+64),((length-1)*64+1):((length-1)*64+64));%行-->--,列-->--
    end
end
```

```

        end
    end
8. matrix1=block_eight(:, :, 10)      %取出第十块宏块
9. fan_qt=blockproc(matrix1,[8,8],@(block_struct)block_struct.data.*Y_Table);
                                     %反量化
10. fun4 = @(block_struct) idct2(block_struct.data);%定义句柄函数，进行逆 DCT 变换
    I2 = blockproc(fan_qt,[8, 8],fun4);    %调用函数
11. I3=uint8(I2+128);                  %由于 dct 量化中减了 128,这里加上 128
12. imshow(I3); figure;                %显示恢复的宏块空域图像

```

2) 实验结果

1. 恢复的第 10 个宏块对应的原图像



图 7 原图对应的宏块



图 8 恢复的宏块

2.分析 JPEG 压缩所引起的块效应

在图象的强边缘处有不连续的现象出现，这是由于在图象的强边缘处出现。在低码率下，DCT 的很多高频系数被量化为零，结果与强边缘有关的高频分量在变换域内不能完全被体现。另外，还有可能使重建图象中块边界处出现亮度突变，这是由于变换域内，直流分量 DC 系数体现了图象块的平均亮度，所以这个系数包含了图象块的大部分能量。在平坦区域，亮度的变化很小，但是如果有亮度的递增或递减，在量化取整时进行了四舍五入，可能会导致 DC 系数越过相邻量化级的判决门限。

参考文献：

<https://wenku.baidu.com/view/80935e702f3f5727a5e9856a561252d380eb20b0.html>

4. 实验总结

学习过图像处理这门课程，也学过使用 MATLAB 的基本使用，但是要用 MATLAB 真正的实现一些图像处理还是需要花一些时间学习的。学习的重点在于对内置函数的具体作用、用法、内部原理的学习，学习后再进行实验就轻松许多了。

本次实验实现的 JPEG 的压缩、逆压缩以及一些中间步骤的体验，真正体会到了 JPEG 压缩的全过程，并且体会到了压缩的意义，还学到了一些有关块效应的有关知识和 PSNR 的计算公式。并且熟悉了 MATLAB 的使用。