1. 代码设计思路

首先读取一张图片并将其转换为灰度图。由于我们需要将图片分割为8\*8的小块，所以对于像素数量不能被8整除的图片我们将其适当扩展，并对扩展的部分做补零处理。DCT变换的实现与DFT类似，我们先按照下式构造出8\*8的二维DCT变换矩阵



之后将处理过的灰度图像分割为8\*8的小块，将每一小块按相乘，即可得到该图像的分块DCT变换结果。

由于许多DCT系数都是接近0的值，且绝大部分能量均集中在左上角，因此可以只保留较少的DCT系数，而不会对图像重建质量产生重大影响。我们构造了三个8\*8矩阵，其左上角分别有5、10、20个值为1，其余值为0，与DCT后的分块相点乘，以达到保留5、10、20个DCT系数的目的。最后，我们将以上几种方式所得到的矩阵按照进行IDCT变换，得到重构图像，并利用subplot()函数展示结果。

最后我们编写了一个函数以计算重构图像的PSNR。PSNR的定义为

显然若DCT系数保留越多，其重构图像越接近原始图像，PSNR越高。

1. 实验结果

由结果图像可以看出，丢弃部分DCT系数会导致重建图像有质量损失，但图像仍

然清晰可辨。



同时，DCT系数保留的越多，重建图像质量越高，PSNR也越高。当保留前5个系数时PSNR为28.068，保留前10个系数时PSNR为34.0635，保留前20个系数时PSNR为36.3543。PSNR随保留DCT系数的变化图表如下。

1. 学习体会

实际上DCT实现的过程本身不是十分困难，但是在MATLAB编程实现的过程中有许多细节需要注意，例如完成余弦变换与反变换后得到的矩阵，直接作图会发现一片空白，检查后发现需要将结果转换为8为无符号数才可正常显示。计算PSNR的过程中需要调整算式中各变量的数据类型以得到正确的结果。这些都需要仔细认真地思考与耐心的调试程序，才能最后得到较好的结果。