**Project2-频域分析**

一、运行环境

操作系统：Windows11

Python版本：3.8

第三方库：numpy 1.24.2、opencv 4.7.0、matplotlib 3.7.1、scipy 1.10.1

二、实验过程

①离散余弦变换：

a）填充代码：

|  |
| --- |
| dct：  coefs[p, q] += patch[m, n] \* aq \* ap \* np.cos(np.pi \* (2 \* n + 1) \* q / 2 / nc) \* np.cos(np.pi \* (2 \* m + 1) \* p / 2 / nr)  idct：  patch[m, n] += coefs[p, q] \* aq \* ap \* np.cos(np.pi \* (2 \* n + 1) \* q / 2 / nc) \* np.cos(np.pi \* (2 \* m + 1) \* p / 2 / nr) |

b）改变nkeep的数值观察恢复图像的质量：

nkeep=2时，



nkeep=4时，

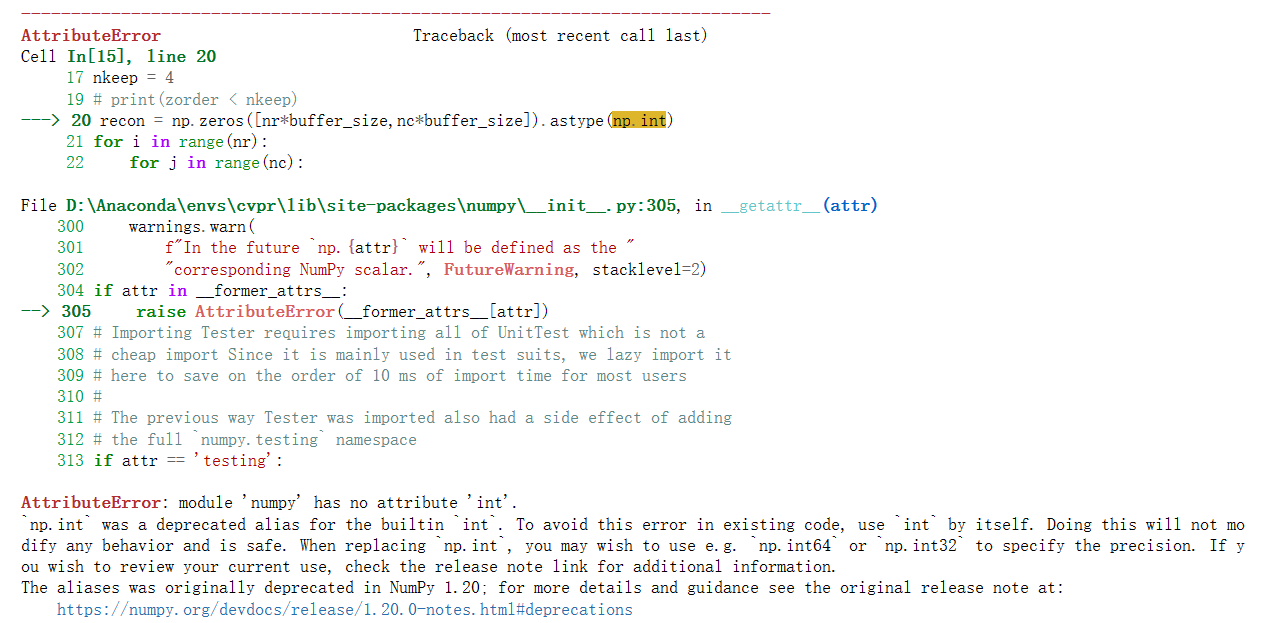


nkeep=8时，



观察发现，随着nkeep的增大，恢复图像的质量变得更清晰。此外，通过本实验可以发现，利用矩阵进行快速离散余弦变换的速度显著高于普通的循环求和计算。

另外，填充完代码后运行报错：



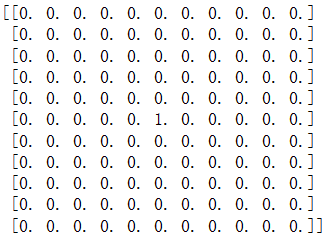
将np.int改为np.int32后代码成功运行。

②滤波器的频谱分析：

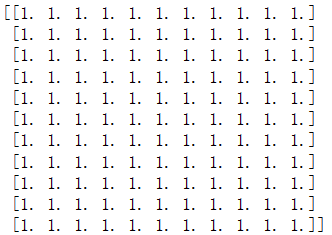
a）填充代码：

|  |
| --- |
| fft：  f1\_fft = np.fft.fft2(f1) f1\_fft = np.fft.fftshift(f1\_fft) f2\_fft = np.fft.fft2(f2) f2\_fft = np.fft.fftshift(f2\_fft)  dct：  f1\_dct = dct(f1, axis=1, norm=**'ortho'**) f1\_dct = dct(f1\_dct, axis=0, norm=**'ortho'**) f2\_dct = dct(f2, axis=1, norm=**'ortho'**) f2\_dct = dct(f2\_dct, axis=0, norm=**'ortho'**) |

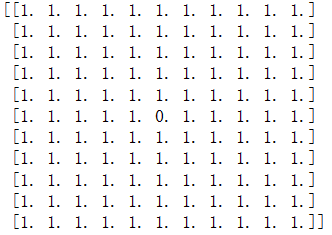
b）傅里叶变换：



f1是一个均值滤波器（低通），故傅里叶变换后其中心为1，代表低频分量通过。

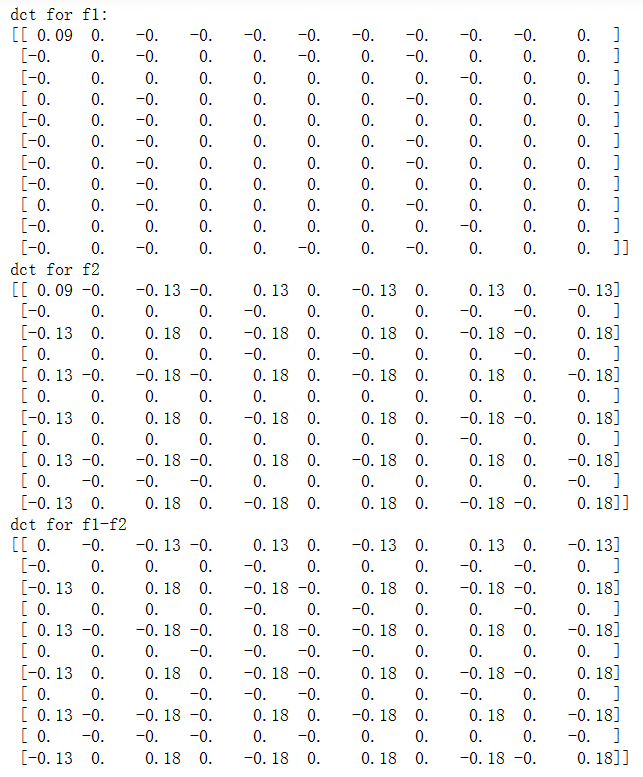


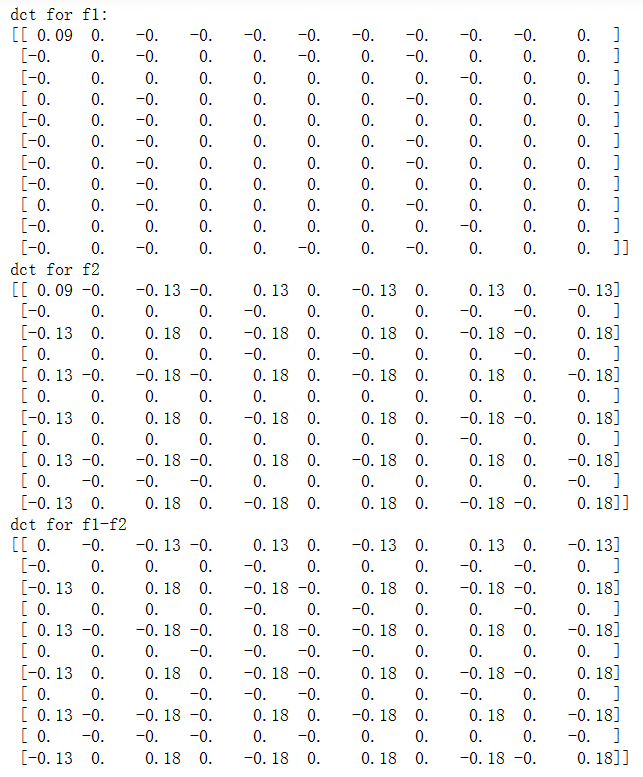
f2是一个冲激响应滤波器（全通），故傅里叶变换后矩阵全1，代表所有频率分量通过。



f1-f2是一个高通滤波器，故傅里叶变换后其中心为0，四周全1，代表高频分量通过。

c）离散余弦变换：





离散余弦变换后的矩阵左上角表示低频分量，右下角表示高频分量。