**基于numpy傅立叶变换的低通滤波**

滤除高频成分，保留低频成分，在频域中实现平滑处理。

**1、代码实现：**

# 基于numpy傅立叶变换的低通滤波

#导入相关库

import cv2

import numpy as np

import matplotlib.pyplot as plt

#使用cv2 读入图片

new\_img=cv2.imread('../image/1.png',0)

#numpy中的傅立叶变化

f1=np.fft.fft2(new\_img)

f1\_shift=np.fft.fftshift(f1)

#np.fft.fftshift()函数来实现平移,让直流分量在输出图像的重心

rows,cols=new\_img.shape

crow,ccol=int(rows/2),int(cols/2) #计算频谱中心

mask=np.zeros((rows,cols),np.uint8) #生成rows行cols的矩阵，数据格式为uint8

mask[crow-30:crow+30,ccol-30:ccol+30]=1 #将靠近频谱中心的部分低通信息 设置为1，属于低通滤波

f1\_shift=f1\_shift\*mask

#傅立叶逆变换

f\_ishift=np.fft.ifftshift(f1\_shift)

img\_back=np.fft.ifft2(f\_ishift)

img\_back=np.abs(img\_back)

img\_back=(img\_back-np.amin(img\_back))/(np.amax(img\_back)-np.amin(img\_back))

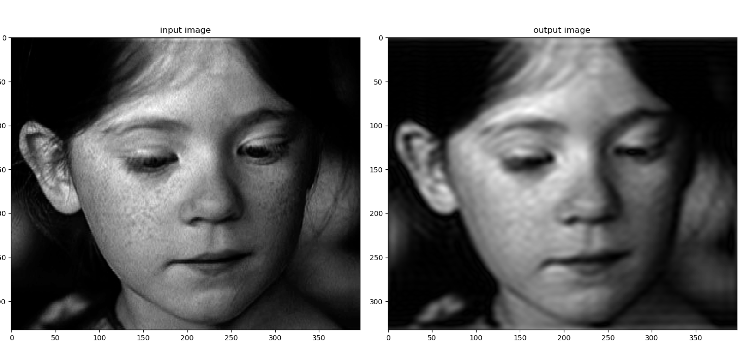
plt.figure(figsize=(15,15))

plt.subplot(121),plt.imshow(new\_img,cmap='gray'),plt.title('input image')

plt.subplot(122),plt.imshow(img\_back,cmap='gray'),plt.title('output image')

plt.show()

**2、实验结果：**



原始图像与低通滤波后图像