Họ và tên: Trần Gia Long Môn: Xử lý ảnh

MSSV: N20DCCN035

Lớp: D20CQCNPM01-N

**Homework 5**

**Bài làm**

**Phần 1:**

**Câu a:**

Khai báo thư viện:

import numpy as np

import matplotlib.pylab as plt

Tạo hàm đọc file và cân bằng ảnh

def read\_file(filename,size):

    img=np.fromfile(filename,dtype=np.uint8)

    img=img.reshape((size,size))

    return img

def full\_scale\_contrast(img):

    min=np.min(img)

    max=np.max(img)

    if min == max:

        return img

    scale=255/(max-min)

    img=(img-min)\*scale

    img=np.round(img) # hoặc img=img.astype(np.uint8)

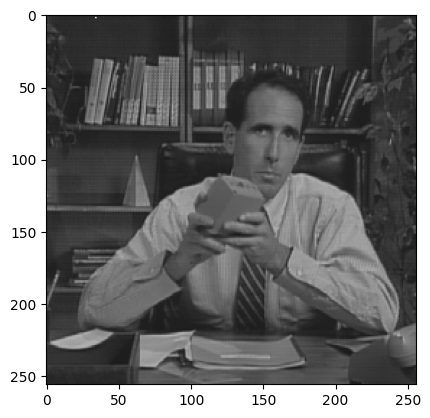
    return img

Hiển thị ảnh đầu

img=read\_file('salesman.bin',256)

print(img.shape)

plt.imshow(img,cmap='gray')



Sử dụng image domain convolution để tạo output image

img2= np.zeros((262,262)) # padding\_image

img2[3:259,3:259]=img # 3 to 258 = 256

img3= np.zeros((262,262))# result

filter = np.ones((7,7))/49

for row in range(3,259):

    for col in range(3,259):

        img3[row,col]=np.sum(img2[row-3:row+4,col-3:col+4]\*filter) # 3-3: 3+4 = 0:6 (range(0,7))

Hiển thị kết quả

plt.figure(figsize=(10,10))

plt.subplot(1,2,1)

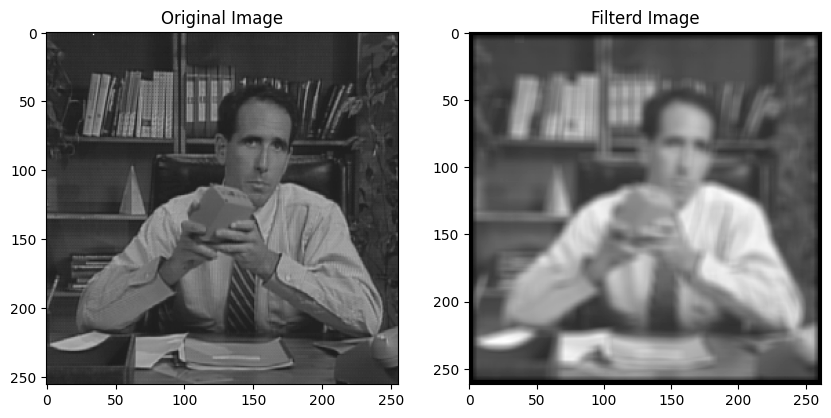
plt.title('Original Image')

plt.imshow(img,cmap='gray')

plt.subplot(1,2,2)

plt.title('Filterd Image')

plt.imshow(full\_scale\_contrast(img3),cmap='gray')



**Câu b:**

Tạo impulse response image

H=np.zeros((128,128)) # impulse response image

center= 128//2

H[center-3:center+4,center-3:center+4]=1/49 # = np.ones((7,7))/49

Khởi tạo kích cỡ phù hợp cho kết quả

zero\_padded\_size=256 + 128 - 1

print(zero\_padded\_size)

Tính các output yêu cầu của đề bài

# zero\_padded\_original\_img

ZPO=np.zeros((zero\_padded\_size,zero\_padded\_size))

ZPO[:256,:256]=img

# zero\_padded\_impulse\_response\_img

ZPI=np.zeros((zero\_padded\_size,zero\_padded\_size))

ZPI[:128,:128]=H

#DFT

ZPOtilde=np.fft.fft2(ZPO)

ZPItilde=np.fft.fft2(ZPI)

#centered DFT log-magnitude spectrum

ZPOtilde\_center=np.log(1+np.abs(np.fft.fftshift(ZPOtilde)))

ZPItilde\_center=np.log(1+np.abs(np.fft.fftshift(ZPItilde)))

#zero padded output image

OItilde= ZPOtilde\* ZPItilde

OI= (np.fft.ifft2(OItilde)).real

#centered DFT log-magnitude spectrum of the zero padded output image

OItilde\_center= np.log(1+np.abs(np.fft.fftshift(OItilde)))

#final 256 × 256 output image

FI=OI[64:320,64:320]

Hiển thị kết quả

plt.figure(figsize=(16,8))

plt.subplot(2,4,1)

plt.title('Origin image')

plt.imshow(full\_scale\_contrast(img),cmap='gray')

plt.subplot(2,4,2)

plt.title('Zero padded original image')

plt.imshow(full\_scale\_contrast(ZPO),cmap='gray')

plt.subplot(2,4,3)

plt.title('Zero padded impulse response image')

plt.imshow(full\_scale\_contrast(ZPI),cmap='gray')

plt.subplot(2,4,4)

plt.title('centered DFT ZPO')

plt.imshow(full\_scale\_contrast(ZPOtilde\_center),cmap='gray')

plt.subplot(2,4,5)

plt.title('centered DFT ZPI')

plt.imshow(full\_scale\_contrast(ZPItilde\_center),cmap='gray')

plt.subplot(2,4,6)

plt.title('centered DFT output')

plt.imshow(full\_scale\_contrast(OItilde\_center),cmap='gray')

plt.subplot(2,4,7)

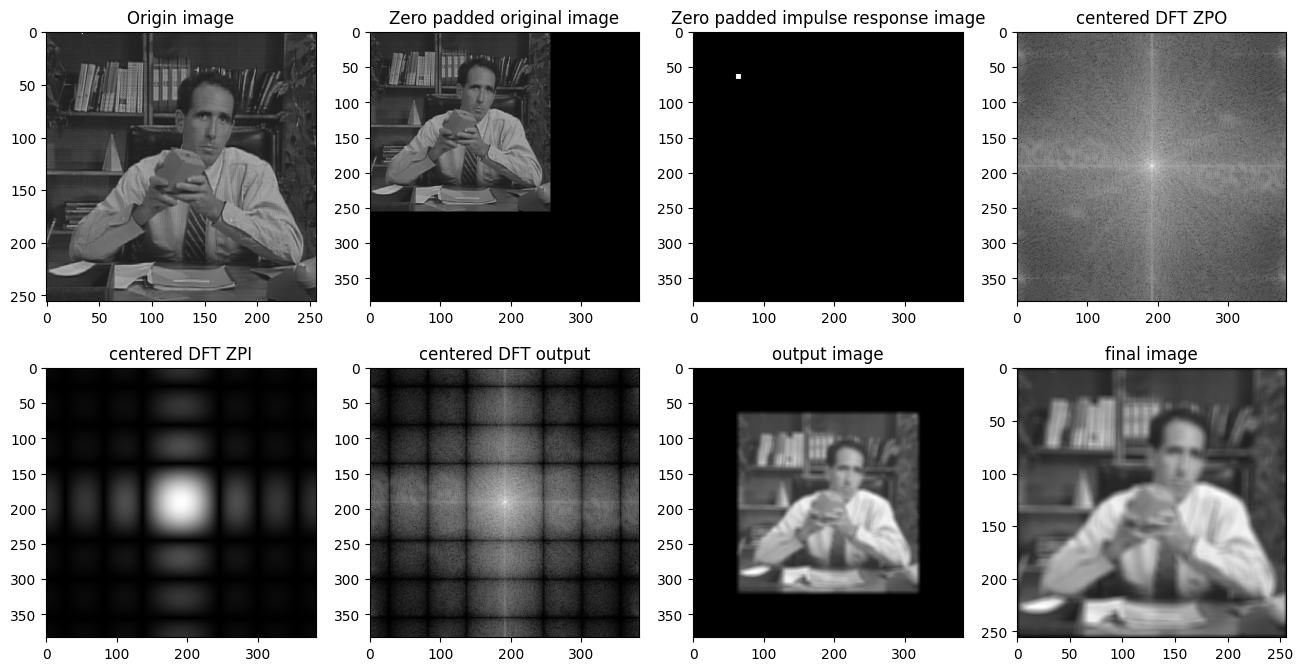
plt.title('output image')

plt.imshow(full\_scale\_contrast(OI),cmap='gray')

plt.subplot(2,4,8)

plt.title('final image')

plt.imshow(full\_scale\_contrast(FI),cmap='gray')



So sánh kết quả với câu a

#Verify that the output image is the same as the one in part (a)

#Y1a=img3, Y1b=FI

Y1a=full\_scale\_contrast(img3[3:262-3,3:262-3])

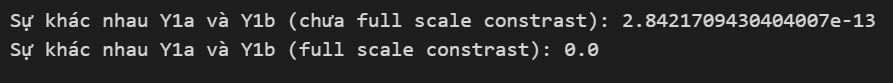
Y1b=full\_scale\_contrast(FI)

max\_difference\_before=np.max(np.abs(FI-img3[3:262-3,3:262-3]))

max\_difference\_after=np.max(np.abs(Y1b-Y1a))

print('Sự khác nhau Y1a và Y1b (chưa full scale constrast): '+str(max\_difference\_before))

print('Sự khác nhau Y1a và Y1b (full scale constrast): '+str(max\_difference\_after))



**Câu c:**

Tính các yêu cầu của bài toán

#zero-phase impulse response image

H= np.zeros((256,256))

H[125:132,125:132]=1/49

H2=np.fft.fftshift(H)

# zero\_padded\_original\_img

ZPO2=np.zeros((512,512))

ZPO2[:256,:256]=img

# zero padded zero-phase impulse response image

ZPI2=np.zeros((512,512))

ZPI2[:128,:128]=H2[:128,:128]

ZPI2[:128,384:512]=H2[:128,128:256]

ZPI2[384:512,:128]=H2[128:256,:128]

ZPI2[384:512,384:512]=H2[128:256,128:256]

#final 256 × 256 output image

FI2=np.fft.ifft2(np.fft.fft2(ZPO2)\*np.fft.fft2(ZPI2))

FI2=FI2.real[:256,:256]

Hiển thị kết quả

plt.figure(figsize=(20,8))

plt.subplot(1,4,1)

plt.title('Origin image')

plt.imshow(full\_scale\_contrast(img),cmap='gray')

plt.subplot(1,4,2)

plt.title('Zero-phase impulse response image')

plt.imshow(full\_scale\_contrast(H2),cmap='gray')

plt.subplot(1,4,3)

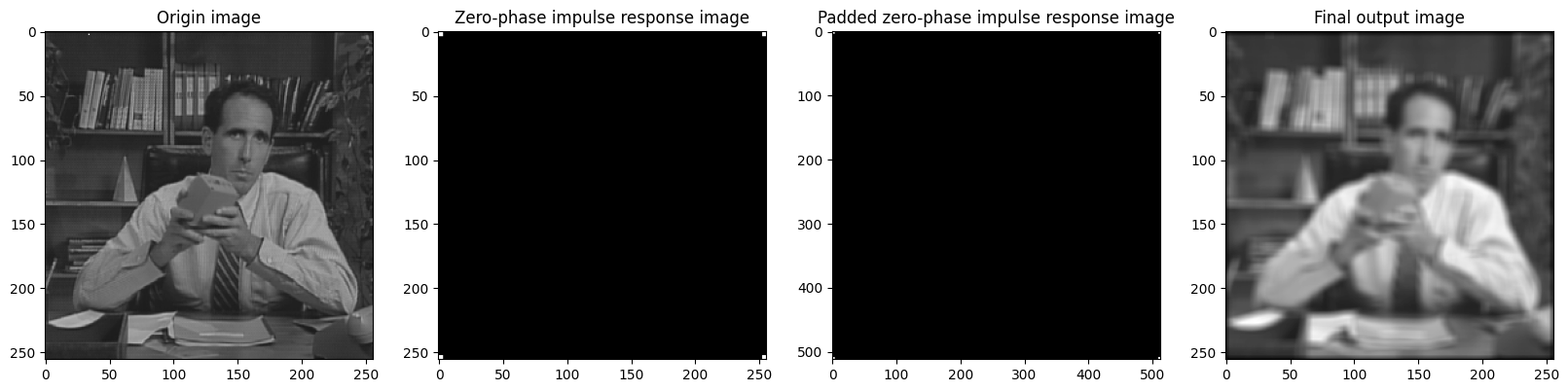
plt.title('Padded zero-phase impulse response image')

plt.imshow(full\_scale\_contrast(ZPI2),cmap='gray')

plt.subplot(1,4,4)

plt.title('Final output image')

plt.imshow(full\_scale\_contrast(FI2),cmap='gray')



So sánh với câu a

#Verify that the output image is the same as the one in part (a)

#Y1a=img3, Y1c=FI2

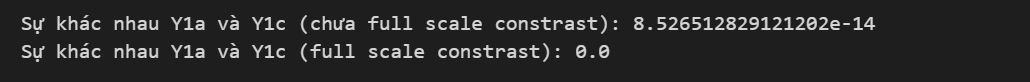
Y1c=full\_scale\_contrast(FI2)

max\_difference\_before=np.max(np.abs(FI2-img3[3:262-3,3:262-3]))

max\_difference\_after=np.max(np.abs(Y1c-Y1a))

print('Sự khác nhau Y1a và Y1c (chưa full scale constrast): '+str(max\_difference\_before))

print('Sự khác nhau Y1a và Y1c (full scale constrast): '+str(max\_difference\_after))



**Phần 2:**

**Câu a:**

Đọc file và tính MSE của 2 trường hợp hi-pass Gaussian và broadband Gaussian so với ảnh gốc

girl=read\_file('girl2.bin',256)

girl\_N=read\_file('girl2Noise32.bin',256)

girl\_NH=read\_file('girl2Noise32Hi.bin',256)

#Hàm tính MSE

def mse(b,a=girl):

    return np.mean((b-a)\*\*2)

mse\_girl\_N=mse(girl\_N,girl)

mse\_girl\_NH=mse(girl\_NH,girl)

Hiển thị kết quả

print('MSE girl2Noise32 & girl2: '+ str(mse\_girl\_N))

print('MSE girl2Noise32Hi & girl2: '+ str(mse\_girl\_NH))

plt.figure(figsize=(16,8))

plt.subplot(1,3,1)

plt.title('Tiffany')

plt.imshow(girl,cmap='gray')

plt.subplot(1,3,2)

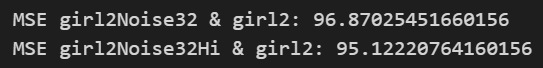
plt.title('Tiffany has broadband Gaussian white noise')

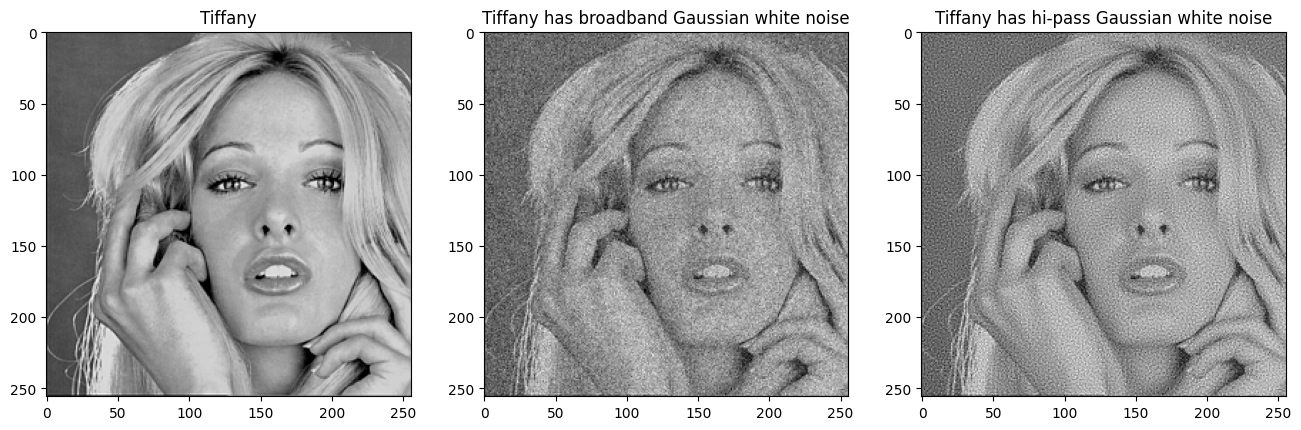
plt.imshow(girl\_N,cmap='gray')

plt.subplot(1,3,3)

plt.title('Tiffany has hi-pass Gaussian white noise')

plt.imshow(girl\_NH,cmap='gray')





**Câu b:**

Tạo hàm low-pass filter áp dụng vào hình ảnh

def apply\_ideal\_lpf(img,cutoff=64):

    # 2 mảng (mảng [-128,127] chạy hàng, chạy cột) (u,v)

    u,v=np.meshgrid(np.arange(-128,128),np.arange(-128,128)) #(-row//2,row//2),(-col//2,col//2)

    hltilde\_center=np.double(np.sqrt(u\*\*2+v\*\*2)<=cutoff)

    hltilde=np.fft.fftshift(hltilde\_center)

    result=np.fft.ifft2(np.fft.fft2(img)\*hltilde).real # real để loại bỏ phức

    return result

Áp dụng filter vào các hình và tính MSE, ISNR

#áp dụng filter

u\_cutoff=64

lpf\_girl=apply\_ideal\_lpf(girl,u\_cutoff) # real để loại bỏ phức

lpf\_girl\_N=apply\_ideal\_lpf(girl\_N,u\_cutoff)

lpf\_girl\_NH=apply\_ideal\_lpf(girl\_NH,u\_cutoff)

#tính MSE cho hình sau khi áp dụng filter (lpf)

mse\_lpf\_girl=mse(lpf\_girl,girl)

mse\_lpf\_girl\_N=mse(lpf\_girl\_N,girl)

mse\_lpf\_girl\_NH=mse(lpf\_girl\_NH,girl)

#tính ISNR

isnr\_girl\_N=10\*np.log10(mse\_girl\_N/mse\_lpf\_girl\_N)

isnr\_girl\_NH=10\*np.log10(mse\_girl\_NH/mse\_lpf\_girl\_NH)

Kết quả MSE, ISNR

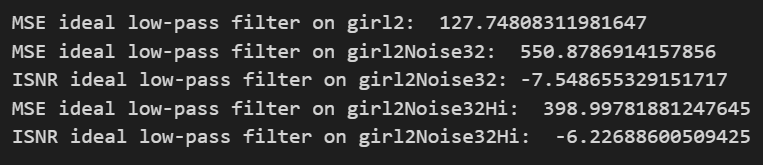
print('MSE ideal low-pass filter on girl2: ',mse\_lpf\_girl)

print('MSE ideal low-pass filter on girl2Noise32: ',mse\_lpf\_girl\_N)

print('ISNR ideal low-pass filter on girl2Noise32:',isnr\_girl\_N)

print('MSE ideal low-pass filter on girl2Noise32Hi: ',mse\_lpf\_girl\_NH)

print('ISNR ideal low-pass filter on girl2Noise32Hi: ',isnr\_girl\_NH)



Hiển thị kết quả hình sau khi áp dụng LPF

plt.figure(figsize=(16,8))

plt.subplot(1,3,1)

plt.title('Tiffany (lpf)')

plt.imshow(full\_scale\_contrast(lpf\_girl),cmap='gray')

plt.subplot(1,3,2)

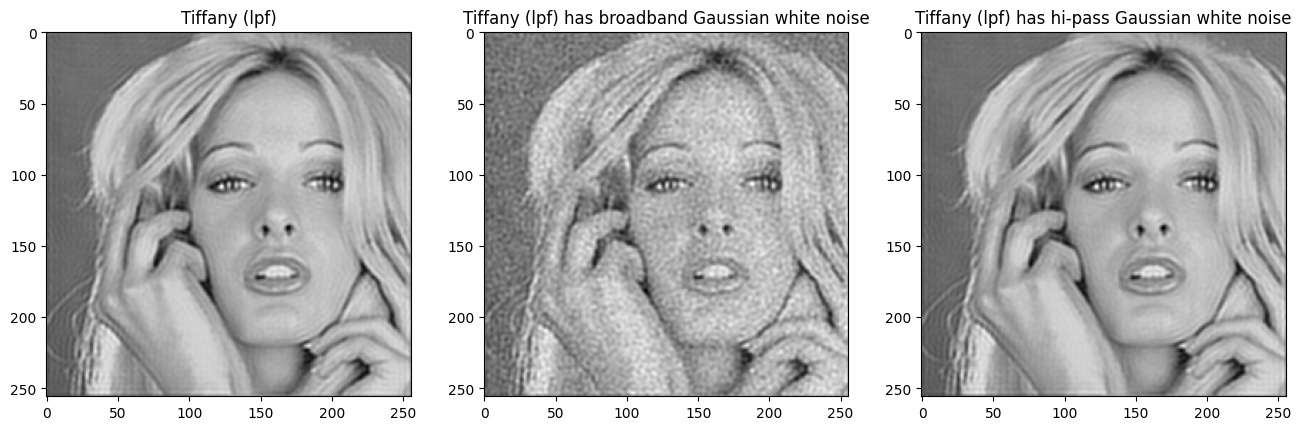
plt.title('Tiffany (lpf) has broadband Gaussian white noise')

plt.imshow(full\_scale\_contrast(lpf\_girl\_N),cmap='gray')

plt.subplot(1,3,3)

plt.title('Tiffany (lpf) has hi-pass Gaussian white noise')

plt.imshow(full\_scale\_contrast(lpf\_girl\_NH),cmap='gray')



**Câu c:**

Tạo hàm áp dụng Gaussian low-pass filter

def apply\_ideal\_lpf\_gaussian(img,cutoff=64):

    sigma\_h=0.19\*256/ cutoff

    u,v=np.meshgrid(np.arange(-128,128),np.arange(-128,128)) #(-row//2,row//2),(-col//2,col//2)

    htilde\_center=np.exp((-2\*np.pi\*\*(2\*sigma\_h\*\*2))/((256\*\*2)\*(u\*\*2+v\*\*2)))

    htilde=np.fft.fftshift(htilde\_center)

    H = np.fft.ifft2(htilde).real

    H2=np.fft.fftshift(H)

    ZPH2=np.zeros((512,512))

    ZPH2[:256,:256]=H2

    ZPO3=np.zeros((512,512))

    ZPO3[:256,:256]=img

    result=np.fft.ifft2(np.fft.fft2(ZPO3)\*np.fft.fft2(ZPH2)).real

    return result[128:384,128:384]

Áp dụng GLPF cho các hình ảnh và tính MSE, ISNR

#Áp dụng vào các hình gaussian lpf

u\_cutoff\_h=64

glpf\_girl=apply\_ideal\_lpf\_gaussian(girl,u\_cutoff\_h)

glpf\_girl\_N=apply\_ideal\_lpf\_gaussian(girl\_N,u\_cutoff\_h)

glpf\_girl\_NH=apply\_ideal\_lpf\_gaussian(girl\_NH,u\_cutoff\_h)

#Tính MSE

mse\_glpf\_girl=mse(glpf\_girl,girl)

mse\_glpf\_girl\_N=mse(glpf\_girl\_N,girl)

mse\_glpf\_girl\_NH=mse(glpf\_girl\_NH,girl)

#Tính ISNR

isnr\_glpf\_girl\_N=10\*np.log10(mse\_girl\_N/mse\_glpf\_girl\_N)

isnr\_glpf\_girl\_NH=10\*np.log10(mse\_girl\_NH/mse\_glpf\_girl\_NH)

Hiển thị kết quả MSE, ISNR

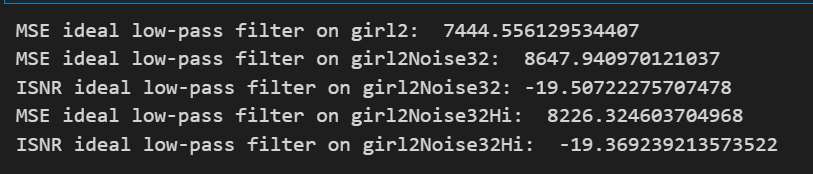
print('MSE ideal low-pass filter on girl2: ',mse\_glpf\_girl)

print('MSE ideal low-pass filter on girl2Noise32: ',mse\_glpf\_girl\_N)

print('ISNR ideal low-pass filter on girl2Noise32:',isnr\_glpf\_girl\_N)

print('MSE ideal low-pass filter on girl2Noise32Hi: ',mse\_glpf\_girl\_NH)

print('ISNR ideal low-pass filter on girl2Noise32Hi: ',isnr\_glpf\_girl\_NH)



Hiển thị hình ảnh đã áp dụng GLPF

plt.figure(figsize=(16,8))

plt.subplot(1,3,1)

plt.title('Tiffany (glpf)')

plt.imshow(full\_scale\_contrast(glpf\_girl),cmap='gray')

plt.subplot(1,3,2)

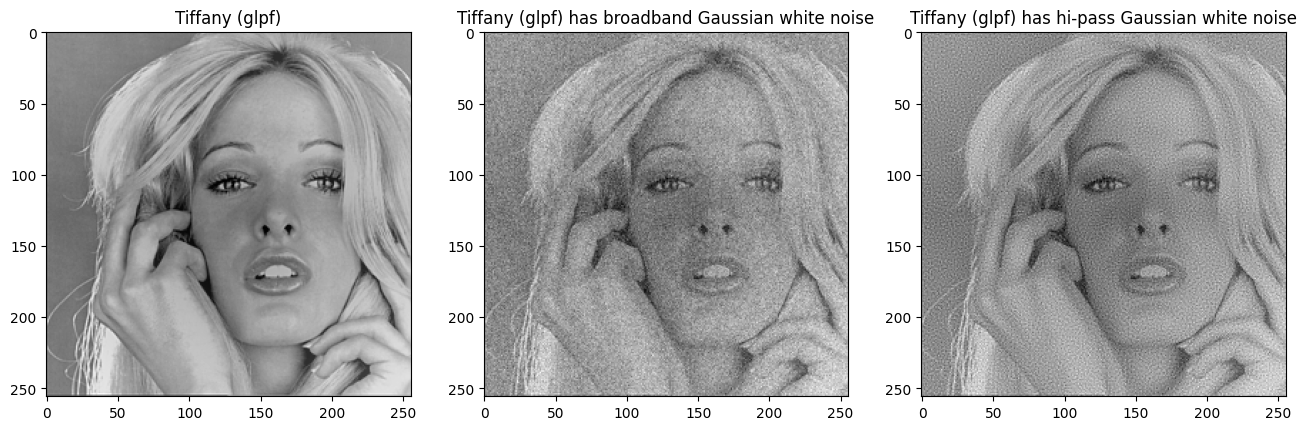
plt.title('Tiffany (glpf) has broadband Gaussian white noise')

plt.imshow(full\_scale\_contrast(glpf\_girl\_N),cmap='gray')

plt.subplot(1,3,3)

plt.title('Tiffany (glpf) has hi-pass Gaussian white noise')

plt.imshow(full\_scale\_contrast(glpf\_girl\_NH),cmap='gray')



Câu d:

Áp dụng GLPF với cutoff = 77.5

#Áp dụng vào các hình gaussian lpf

u\_cutoff\_h=77.5

glpf\_girl2=apply\_ideal\_lpf\_gaussian(girl,u\_cutoff\_h)

glpf\_girl\_N2=apply\_ideal\_lpf\_gaussian(girl\_N,u\_cutoff\_h)

glpf\_girl\_NH2=apply\_ideal\_lpf\_gaussian(girl\_NH,u\_cutoff\_h)

#Tính MSE

mse\_glpf\_girl2=mse(glpf\_girl2,girl)

mse\_glpf\_girl\_N2=mse(glpf\_girl\_N2,girl)

mse\_glpf\_girl\_NH2=mse(glpf\_girl\_NH2,girl)

#Tính ISNR

isnr\_glpf\_girl\_N2=10\*np.log10(mse\_girl\_N/mse\_glpf\_girl\_N2)

isnr\_glpf\_girl\_NH2=10\*np.log10(mse\_girl\_NH/mse\_glpf\_girl\_NH2)

Hiển thị

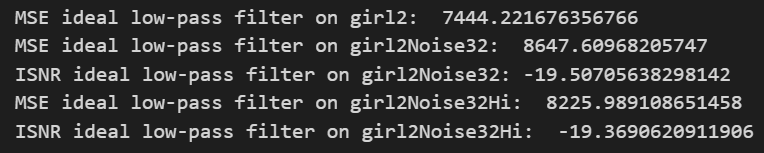
print('MSE ideal low-pass filter on girl2: ',mse\_glpf\_girl2)

print('MSE ideal low-pass filter on girl2Noise32: ',mse\_glpf\_girl\_N2)

print('ISNR ideal low-pass filter on girl2Noise32:',isnr\_glpf\_girl\_N2)

print('MSE ideal low-pass filter on girl2Noise32Hi: ',mse\_glpf\_girl\_NH2)

print('ISNR ideal low-pass filter on girl2Noise32Hi: ',isnr\_glpf\_girl\_NH2)



plt.figure(figsize=(16,8))

plt.subplot(1,3,1)

plt.title('Tiffany (glpf)')

plt.imshow(full\_scale\_contrast(glpf\_girl2),cmap='gray')

plt.subplot(1,3,2)

plt.title('Tiffany (glpf) has broadband Gaussian white noise')

plt.imshow(full\_scale\_contrast(glpf\_girl\_N2),cmap='gray')

plt.subplot(1,3,3)

plt.title('Tiffany (glpf) has hi-pass Gaussian white noise')

plt.imshow(full\_scale\_contrast(glpf\_girl\_NH2),cmap='gray')

