HOMEWORK 5

Họ và tên: Trịnh Thanh Sơn

Lớp: D20CQCNPM02-N

MSSV: N20DCCN134

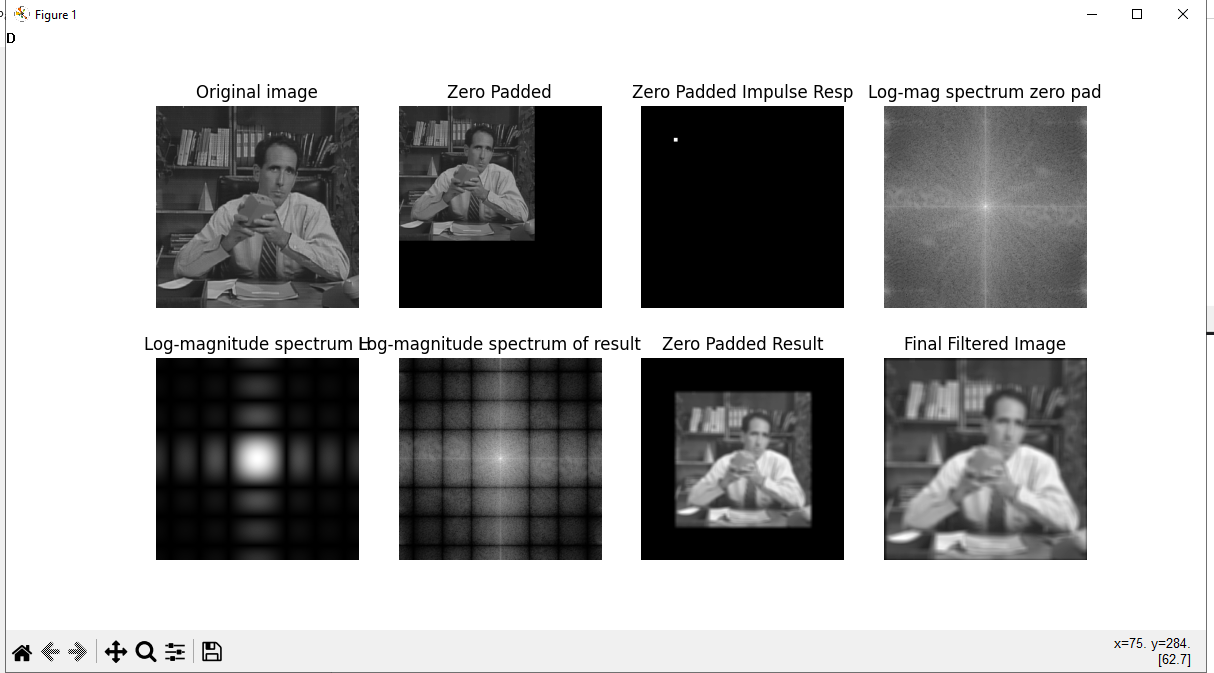
1. **Bài 1:**

* Code:
* import numpy as np  
  import matplotlib.pyplot as plt  
    
  def read\_data(data\_path, size):  
   with open(data\_path, 'rb') as file:  
   data = np.fromfile(file, dtype=np.uint8, count=size\*size)  
   return np.reshape(data, (size, size))  
    
  def stretch(img):  
   return (img - np.min(img)) / (np.max(img) - np.min(img)) \* 255.0  
    
  data\_path = 'dataset/salesman.sec'  
  size = 256  
  read = read\_data(data\_path, size)  
  plt.figure(figsize=(12, 6))  
  plt.subplot(1, 2, 1)  
  plt.imshow(read, cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('Original Image', fontsize=18)  
  X = np.zeros((262, 262))  
  X[4:260, 4:260] = read  
  Y = np.zeros((262, 262))  
  for row in range(4, 261):  
   for col in range(4, 261):  
   Y[row, col] = np.sum(X[row-3:row+4, col-3:col+4]) / 49  
  Y2 = stretch(Y[4:260, 4:260])  
  plt.subplot(1,2, 2)  
  plt.imshow(Y2, cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('Filtered Image', fontsize=18)  
  plt.show()  
    
  #Cau b  
  plt.figure(figsize=(12, 6))  
  plt.subplot(2, 4, 1)  
  plt.imshow(read, cmap='gray')  
  plt.title('Original image', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  size1 = 256 + 128 - 1  
  pX = np.zeros((size1, size1))  
  pX[:256, :256] = read  
  plt.subplot(2, 4, 2)  
  plt.imshow(pX, cmap='gray')  
  plt.title('Zero Padded', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  H = np.zeros((128, 128))  
  H[62:69, 62:69] = 1 / 49  
  pH = np.zeros((size1, size1))  
  pH[:128, :128] = H  
  plt.subplot(2, 4, 3)  
  plt.imshow(pH, cmap='gray')  
  plt.title('Zero Padded Impulse Resp', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  pXtilde = np.fft.fft2(pX)  
  pHtilde = np.fft.fft2(pH)  
  pXtildeDisplay = np.log(1 + np.abs(np.fft.fftshift(pXtilde)))  
  plt.subplot(2, 4, 4)  
  plt.imshow(pXtildeDisplay, cmap='gray')  
  plt.title('Log-mag spectrum zero pad', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  pHtildeDisplay = np.log(1 + np.abs(np.fft.fftshift(pHtilde)))  
  plt.subplot(2, 4, 5)  
  plt.imshow(pHtildeDisplay, cmap='gray')  
  plt.title('Log-magnitude spectrum H', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  pYtilde = pXtilde \* pHtilde  
  pY = np.fft.ifft2(pYtilde)  
  pYtildeDisplay = np.log(1 + np.abs(np.fft.fftshift(pYtilde)))  
  plt.subplot(2, 4, 6)  
  plt.imshow(pYtildeDisplay, cmap='gray')  
  plt.title('Log-magnitude spectrum of result', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  plt.subplot(2, 4, 7)  
  plt.imshow(np.real(pY), cmap='gray')  
  plt.title('Zero Padded Result', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  Y = np.real(pY[64:320, 64:320])  
  plt.subplot(2, 4, 8)  
  plt.imshow(Y, cmap='gray')  
  plt.title('Final Filtered Image', fontsize=12)  
  plt.axis('image')  
  plt.axis('off')  
  plt.show()  
    
  #Cau c  
  def stretch(x):  
   xMax = np.max(x)  
   xMin = np.min(x)  
   scale = 255.0 / (xMax - xMin)  
   y = np.round((x - xMin) \* scale)  
   return y.astype(np.uint8)  
  H1 = np.zeros((256, 256))  
  H1[126:133, 126:133] = 1/49  
  H2 = np.fft.fftshift(H1)  
  print(H2.shape)  
  plt.figure(figsize=(10, 6))  
  plt.subplot(2, 2, 1)  
  plt.imshow(stretch(read), cmap='gray')  
  plt.title('Zero Phase Impulse Resp', fontsize=18)  
  plt.axis('image')  
  plt.axis('off')  
  plt.subplot(2, 2, 2)  
  plt.imshow(stretch(H2), cmap='gray')  
  plt.title('Zero Phase Impulse Resp', fontsize=18)  
  plt.axis('image')  
  plt.axis('off')  
  pX = np.zeros((512, 512))  
  pX[:256, :256] = read  
  pH2 = np.zeros((512, 512))  
  pH2[:128, :128] = H2[:128, :128]  
  pH2[:128, 385:512] = H2[:128, 129:256]  
  pH2[385:512, :128] = H2[129:256, :128]  
  pH2[385:512, 385:512] = H2[129:256, 129:256]  
  plt.subplot(2, 2, 3)  
  plt.imshow(stretch(pH2), cmap='gray')  
  plt.title('Zero Padded zero-phase H', fontsize=18)  
  plt.axis('image')  
  plt.axis('off')  
  Y = np.fft.ifft2(np.fft.fft2(pX) \* np.fft.fft2(pH2))  
  Y = stretch(Y[:256, :256])  
  plt.subplot(2, 2, 4)  
  plt.imshow(Y, cmap='gray')  
  plt.title('Final Filtered Image', fontsize=18)  
  plt.axis('image')  
  plt.axis('off')  
  plt.show()
* Kết quả:

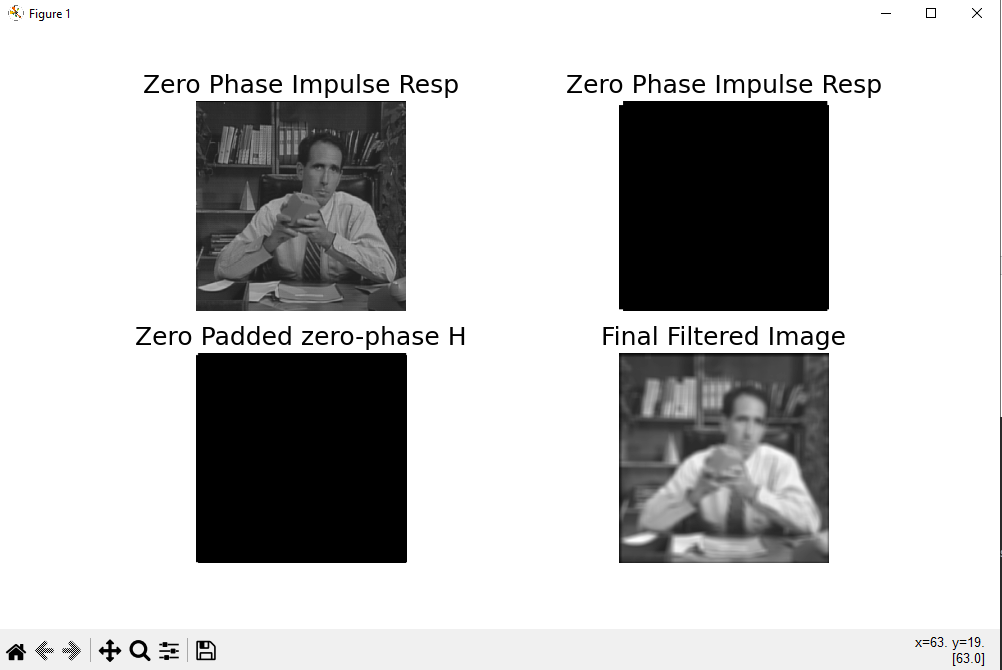
Câu a:



Câu b:



Câu c:



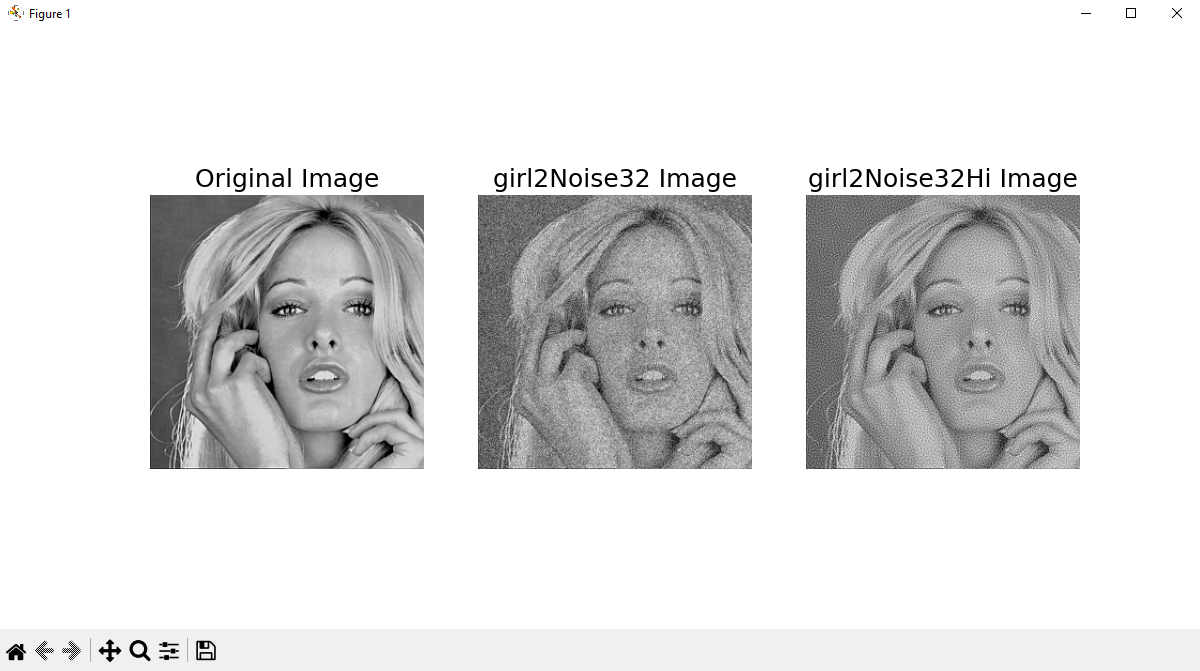
1. **Bài 2:**

* Code:
* import numpy as np  
  import matplotlib.pyplot as plt  
    
  def read\_data(file\_path, size):  
   with open(file\_path, 'rb') as file:  
   data = np.fromfile(file, dtype=np.uint8, count=size\*size)  
   return np.reshape(data, (size, size))  
  X = read\_data('dataset/girl2.sec', 256)  
  X1 = read\_data('dataset/girl2Noise32.sec', 256)  
  X2 = read\_data('dataset/girl2Noise32Hi.sec', 256)  
  Y1 = np.sum((X.astype("float") - X1.astype("float"))\*\*2)  
  Y1 /= float(X.shape[0] \* X1.shape[1])  
  Y2 = np.sum((X.astype("float") - X2.astype(float))\*\*2)  
  Y2 /= float(X.shape[0] \* X2.shape[1])  
  print("Mean Squared Error between girl2 and girl2Noise32 = ", Y1)  
  print("Mean Squared Error between girl2 and girl2Noise32Hi = ", Y2)  
  plt.figure(figsize=(12, 6))  
  plt.subplot(1, 3, 1)  
  plt.imshow(X, cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('Original Image', fontsize=18)  
    
  plt.subplot(1, 3, 2)  
  plt.imshow(X1, cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('girl2Noise32 Image', fontsize=18)  
    
  plt.subplot(1, 3, 3)  
  plt.imshow(X2, cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('girl2Noise32Hi Image', fontsize=18)  
  plt.show()  
    
  #Cau b  
  def stretch2(img):  
   img\_min = np.min(img)  
   img\_max = np.max(img)  
   stretched\_img = 255 \* (img - img\_min) / (img\_max - img\_min)  
   return stretched\_img.astype(np.uint8)  
    
  U\_cutoff = 64  
  [U, V] = np.meshgrid(np.arange(-128, 128), np.arange(-128, 128))  
  HLtildeCenter = np.double(np.sqrt(U\*\*2 + V\*\*2) <= U\_cutoff)  
  HLtilde = np.fft.fftshift(HLtildeCenter)  
  Z = np.fft.ifft2(np.fft.fft2(X) \* HLtilde)  
  Z1 = np.fft.ifft2(np.fft.fft2(X1) \* HLtilde)  
  Z2 = np.fft.ifft2(np.fft.fft2(X2) \* HLtilde)  
  plt.figure(figsize=(12, 6))  
  plt.subplot(1, 3, 1)  
  plt.imshow(stretch2(Z), cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('Original Image', fontsize=18)  
    
  plt.subplot(1, 3, 2)  
  plt.imshow(stretch2(Z1), cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('girl2Noise32 Image', fontsize=18)  
    
  plt.subplot(1, 3, 3)  
  plt.imshow(stretch2(Z2), cmap='gray', vmin=0, vmax=255)  
  plt.axis('image')  
  plt.axis('off')  
  plt.title('girl2Noise32Hi Image', fontsize=18)  
  plt.show()  
    
  Z3 = np.sum(np.abs(X.astype("complex") - Z.astype("complex")) \*\* 2)  
  Z3 /= float(X.shape[0] \* Z.shape[1])  
  Z4 = np.sum(np.abs(X.astype("complex") - Z1.astype("complex")) \*\* 2)  
  Z4 /= float(X.shape[0] \* Z1.shape[1])  
  Z5 = np.sum(np.abs(X.astype("complex") - Z2.astype("complex")) \*\* 2)  
  Z5 /= float(X.shape[0] \* Z2.shape[1])  
  ISNR1 = 10\*np.log10(Y1/Z4)  
  ISNR2 = 10\*np.log10(Y2/Z5)  
  print('MSE of Z', Z3)  
  print('MSE of Z1', Z4)  
  print('MSE of Z2', Z5)  
  print('ISNR of girl2Noise32Hibin = ', ISNR2)  
  print('ISNR of girl2Noise32bin = ', ISNR1)  
    
  #Câu b vs Cau c  
  def GaussianLPF(U\_cutoff\_H, X, X1, X2):  
   SigmaH = 0.19 \* 256 / U\_cutoff\_H  
   U, V = np.meshgrid(np.arange(-128, 128), np.arange(-128, 128))  
   HtildeCenter = np.exp((-2 \* np.pi \*\* 2 \* SigmaH \*\* 2) / (256 \*\* 2) \* (U \*\* 2 + V \*\* 2))  
   Htilde = np.fft.fftshift(HtildeCenter)  
   H = np.fft.ifft2(Htilde)  
   H2 = np.fft.fftshift(H)  
   ZPH2 = np.zeros((512, 512))  
   ZPH2[:256, :256] = H2  
   ZPX = np.zeros((512, 512))  
   ZPX[:256, :256] = X  
   yy = np.fft.ifft2(np.fft.fft2(ZPX) \* np.fft.fft2(ZPH2))  
   T = yy[128:384, 128:384]  
   ZPX[:256, :256] = X1  
   yy = np.fft.ifft2(np.fft.fft2(ZPX) \* np.fft.fft2(ZPH2))  
   T1 = yy[128:384, 128:384]  
   ZPX[:256, :256] = X2  
   yy = np.fft.ifft2(np.fft.fft2(ZPX) \* np.fft.fft2(ZPH2))  
   T2 = yy[128:384, 128:384]  
   ET = np.sum(np.abs(X.astype("complex") - T.astype("complex")) \*\* 2)  
   ET /= float(X.shape[0] \* T.shape[1])  
   ET1 = np.sum(np.abs(X.astype("complex") - T1.astype("complex")) \*\* 2)  
   ET1 /= float(X.shape[0] \* T1.shape[1])  
   ET2 = np.sum(np.abs(X.astype("complex") - T.astype("complex")) \*\* 2)  
   ET2 /= float(X.shape[0] \* T2.shape[1])  
   ISNR3 = 10 \* np.log10(Y1 / ET1)  
   ISNR4 = 10 \* np.log10(Y2 / ET2)  
   print('MSE of girl2bin', ET)  
   print('MSE of girl2Noise32bin', ET1)  
   print('MSE of girl2Noise32Hibin', ET2)  
   print('ISNR of girl2Noise32Hibin = ', ISNR3)  
   print('ISNR of girl2Noise32bin = ', ISNR4)  
   plt.figure(figsize=(12, 6))  
   plt.subplot(1, 3, 1)  
   plt.imshow(stretch2(T), cmap='gray', vmin=0, vmax=255)  
   plt.axis('image')  
   plt.axis('off')  
   plt.title('Original Image', fontsize=18)  
    
   plt.subplot(1, 3, 2)  
   plt.imshow(stretch2(T1), cmap='gray', vmin=0, vmax=255)  
   plt.axis('image')  
   plt.axis('off')  
   plt.title('girl2Noise32 Image', fontsize=18)  
    
   plt.subplot(1, 3, 3)  
   plt.imshow(stretch2(T2), cmap='gray', vmin=0, vmax=255)  
   plt.axis('image')  
   plt.axis('off')  
   plt.title('girl2Noise32Hi Image', fontsize=18)  
   plt.show()  
  GaussianLPF(64, X, X1, X2)  
  GaussianLPF(77.5, X, X1, X2)
* Kết quả:

Câu a:

Mean Squared Error between girl2 and girl2Noise32 = 744.4679107666016

Mean Squared Error between girl2 and girl2Noise32Hi =692.5050201416016



Câu b:

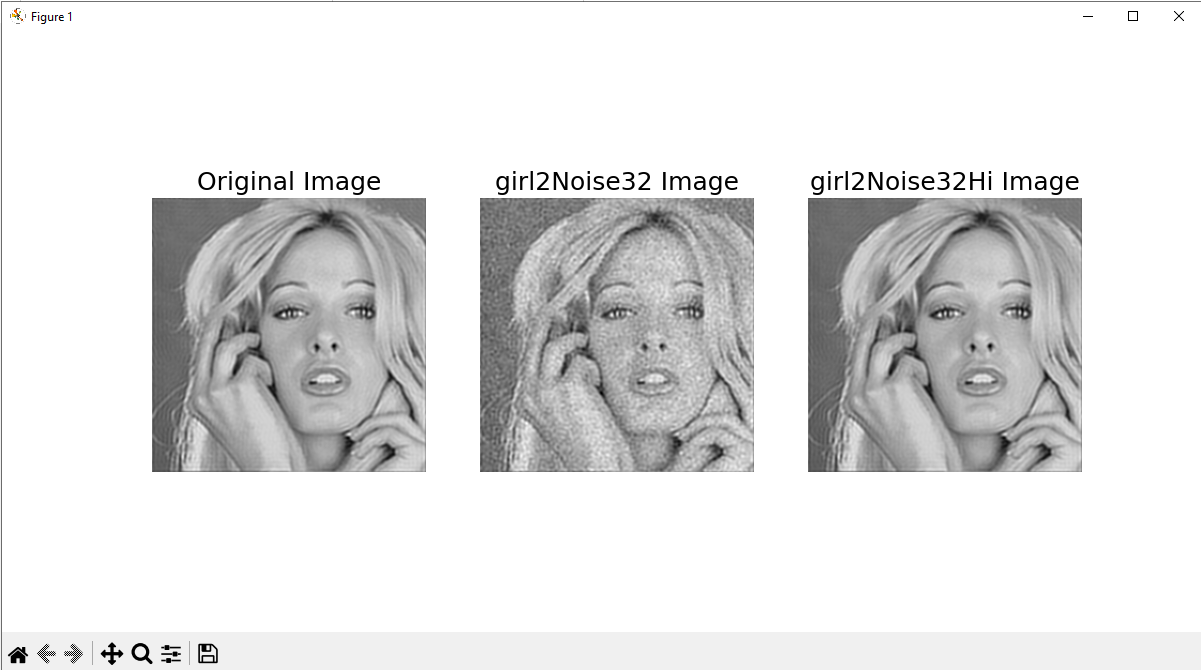
MSE of Z 127.74808311981647

MSE of Z1 550.8786914157856

MSE of Z2 398.99781881247645

ISNR of girl2Noise32Hibin = 2.3945240450623864

ISNR of girl2Noise32bin = 1.3079000912650378



* Câu c:



MSE of girl2bin 91.2584890670409

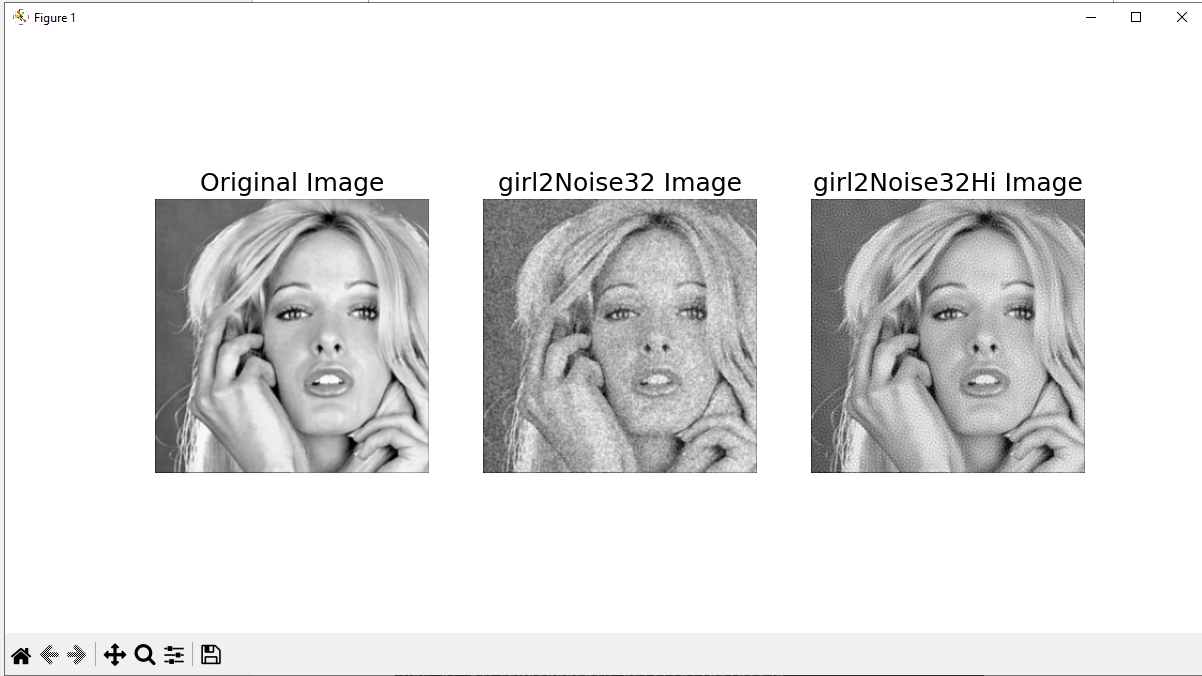
MSE of girl2Noise32bin 534.7740670560556

MSE of girl2Noise32Hibin 91.2584890670409

ISNR of girl2Noise32Hibin = 1.4367564410353562

ISNR of girl2Noise32bin = 8.801496520134169

Câu d:



MSE of girl2bin 60.618360263318024

MSE of girl2Noise32bin 527.7709981931146

MSE of girl2Noise32Hibin 60.618360263318024

ISNR of girl2Noise32Hibin = 1.494004614365322

ISNR of girl2Noise32bin = 10.578187416040636