Snow day Exercise, Feb 20, 2019

Due: End of day the day, Sunday, Feb 24, 2019.

Read about discrete Fourier transform in Numpy

hnttps://docs.scipy.org/doc/numpy-1.13.0/reference/routines.fft.html

Basically:

# FFT in Numpy

f = np.fft.fft2(img)

fshift = np.fft.fftshift(f)

magnitude\_spectrum = 30\*np.log(np.abs(fshift))

phase = np.angle(fshift)

# Inverse FFT

rows, cols = img.shape

crow, ccol = rows/2 , cols/2 # center of the image

fshift[crow-5:crow+5, ccol-5:ccol+5] = 0

f\_ishift = np.fft.ifftshift(fshift)

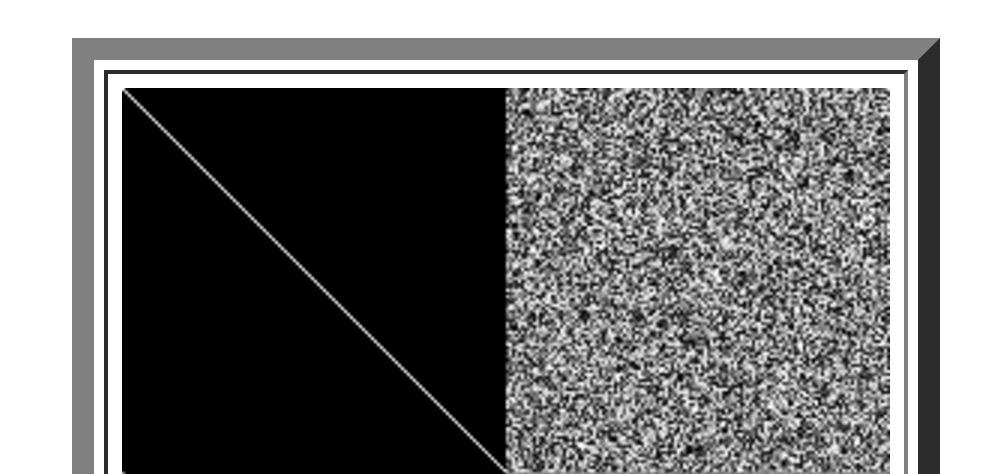
img\_back = np.fft.ifft2(f\_ishift)

1. Open an image (e.g. peppers256.png). Add both Salt and pepper noise, Gaussian noise.

Display FFT images of the original and noisy image

You can follow also the instructions in Lecture\_DenoiseExercise.py

1. What is the FFT of the following images: Using Numpy to find out?



The above image can be generated by the following code:

random\_img = np.random.random((256,256))

Z = np.random.random((256,256)) # Test data