Homework 4 – 100pts

Image Restoration

1) **Create degraded image (30pts)**:
   - ***a.*** [15pts] Apply 5×5 uniform out-of-focus blur to the input image provided, $s(n_1,n_2)$. Display the images and compare with the original to see the degradation. Describe the effects on specific image content. Recover the original image by inverse filtering in the Fourier domain. (Y is the DFT of the blurred image, H(f1,f2) is the DFT of the blur h):

$$\hat{S}(f_1, f_2) = \frac{Y(f_1, f_2)}{H(f_1, f_2)}$$

     Why is it possible to recover exactly the original image?
   - b. [15pts] Then, using function *imnoise()*, add white Gaussian noise with 30 dB SNR to the blurred image to obtain the new noisy and blurred image $y(n_1,n_2)$. Compute PSNR of the blurred & noisy image first using the original image as reference, and then using the blurred image as reference using Matlab function *psnr*(A,ref). Which PSNR is higher and why? Is the PSNR close to the SNR of the noise?

Hints for Q1a

- For blur: create the average 5x5 filter using h = fspecial ('average', 5)
- to compute the args of function *imnoise: noise mean = 0, noise variance is calculated as follows:*

$$SNR = 10 \cdot log_{10} \frac{P_{ss}}{P_{vv}}$$

`The numerator is the power of the signal, which can be calculated as rms^2 using MATLAB rms(), or write your own code. The denominator is the power of the noise, or simply the variance of the noise. If SNR must be 30, then solve for variance of the noise in the above equation. The denominator is the variance of the noise.`

`You can use y = awgn(x,snr,signalpower)with args snr=30 and signalpower = 'measured' but you will get a few points less for using the simpler method.`

Hints for Q1b

- Remember to use H = fft2(h, size(image,1),size(image,2)) so the size of H matches the size of Y

2) **[30pts] Inverse filter on the blurred-noisy image**: Apply the inverse filter in the DFT domain to restore the original image from $y(n_1,n_2)$ by computing an estimate of image $s(n_1,n_2)$ from the image $y(n_1,n_2)$, i.e. the blurred, noisy image from Q1b:

Then, inverse filtering can be expressed in the frequency domain as

$$\hat{S}(f_1, f_2) = \frac{Y(f_1, f_2)}{H(f_1, f_2)}$$

Hint1: as before, H(f1,f2) is the DFT of the blur h, inverse filter neglects the noise term. Also, H = fft2(h, size(image,1),size(image,2))
Hint2: Y=fft2(y).
Hint3: to do inverse fft use ifft2, and to avoid zeros in H, add a small constant value, start at 0.001 and find a constant that will recover a reasonable image. See examples of restoration in https://blogs.mathworks.com/steve/2007/08/13/image-deblurring-introduction/

3) **[40pts] Apply the CLS filter in the DFT domain to restore the original image.**

$$\hat{S}(u_1, u_2) = \frac{H^*(u_1, u_2)}{|H(u_1, u_2)|^2 + \alpha |L(u_1, u_2)|^2} Y(u_1, u_2)$$

where $L(u_1, u_2)$ denotes the eigenvalues of the regularization operator $\mathbf{L}$.

How do you handle image borders? How does the result change for different choices of $|L(u_1,u_2)|^2$ and different values of α? Try at least two different $|L(u_1,u_2)|^2$ and three different values of α for each $|L(u_1,u_2)|^2$.
Hint: Different choices for L: L is a laplacian kernel which can be generated with fun fspecial, use arg alpha values in the range [0, 1], default is 0.2. NOTE: the alpha in fspecial-laplacian is not the same as the alpha in the formula above.
try different choices, and then transformed via fft2 as you did before for h making sure the size is same as y.
Hint: **for the alpha in the formula**: check for values around 0.05, notice that lower values make the image brighter.

Report and code: Not a long report, but answer and comment on all the items required above. It is compulsory to upload the PDF version of the report AND the code (Matlab or Python).