2110431 Introduction to Digital Imaging

2147329 Digital Image Processing and Vision Systems

Homework #2

Deadline: October 10, 2023 @23:59

Submissions: only PDF version of this file

Use these commands in colab to download the images.

```
!wget https://drive.google.com/uc?id=1dhh4m9VRLUSmbaHfge2iRSW5Azkpefco  -O kitty3.png
!wget https://drive.google.com/uc?id=1o0UMPTyUFzX9CaQp-BwYXgkCho1Zo6yL  -O kitty55.png
!wget https://drive.google.com/uc?id=1Jk0cEtQt4HxkLcKlmTHukpb22gJZ4dmL -O noisy_kitty55.png
!wget https://drive.google.com/uc?id=1xCNA5338nzj1GgGQ5-oBA1dKCW1Mvn21 -O hillbefore_noise10%.jpg
```

1. Apply ideal low pass filter in frequency domain on "Kitty3.png" image which has $M \, x \, N$ pixels. Find the minimum cutoff frequency (C) in integer that still maintain the total image power P_T more than 99%. Where the total image power, P_T is calculated by summing the components of spectrum power at each point (u, v), for $u = 1, 2, ... M - 1$ and $v = 1, 2, ..., N - 1$

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u, v)$$

$P(u, v)$ is the spectrum power provided in the lecture slides

α percent of the image power can be calculated from 100 x P_{T_f} / P_{T_org}, where P_{T_org} is the total image power of the original image and P_{T_f} of the filtered image

Put your results in the blank box below

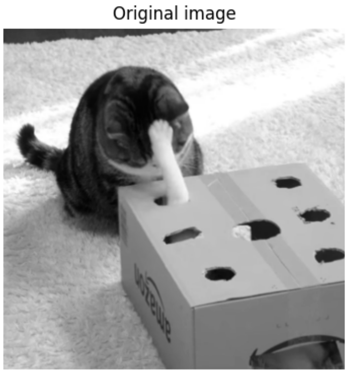
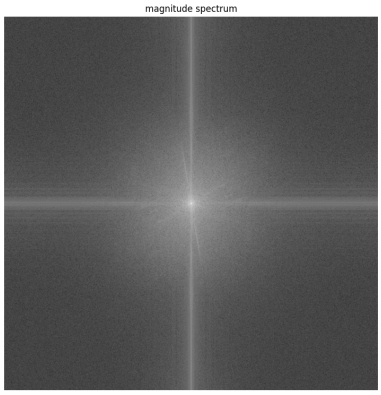
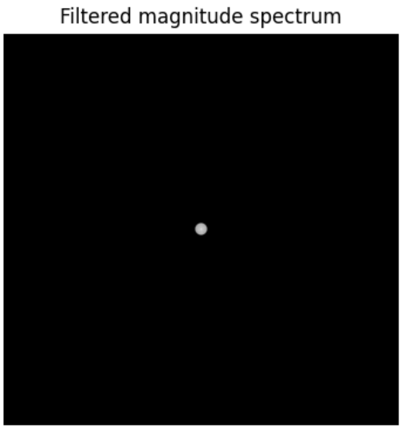
Cutoff frequency (D0) =

16

α =

99.04

Original Image ("kitty55.png")	Fourier Spectrum of the original image

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Fourier Spectrum of the filtered image</td>
<td>Filtered images (P_T > 99%)</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
</tr>
</table>

Put your code here:

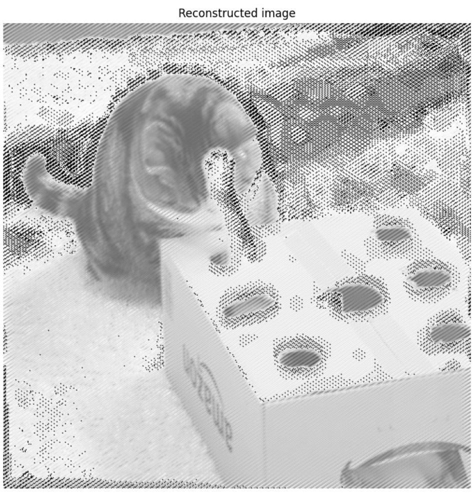
(you can paste all the code or put only the highlighted code for this problem)

```
target_d0 = None
target_filter = None
target_filtered_image = None
target_filtered_power_map = None
target_filtered_power = None
target_power_ratio = None

for D0 in range(max(KITTY.shape),0, -1):
filter = ILPF(KITTY.shape[0],
```

```
KITTY.shape[1],
cut_off_freq=D0)
filtered_image = kitty_fft_shifted * filter

filtered_power_map = np.real(filtered_image)**2 + np.imag(filtered_image)**2
filtered_power = np.sum(filtered_power_map)

Power_ratio = round(filtered_power/KITTY_POWER*100,2)

print(f'Current D0 = {D0}\t|Total power = {filtered_power}\t| Power Ratio =
{Power_ratio} %')
if Power_ratio < 99:
print(f'Done!!! Minimum D0 is {D0+1}.')
target_d0 = D0+1
break

target_filter = ILPF(KITTY.shape[0],
KITTY.shape[1],
cut_off_freq=target_d0)

target_filtered_image = kitty_fft_shifted * filter
target_filtered_magnitude_spectrum = np.log(1 + np.abs(target_filtered_image))

target_filtered_image_ishifted =np.fft.ifftshift(target_filtered_image)
target_filtered_inverse = np.fft.ifft2(target_filtered_image_ishifted)

target_filtered_power_map = np.real(filtered_image)**2 +
np.imag(filtered_image)**2
target_filtered_power = np.sum(filtered_power_map)

target_power_ratio = round(filtered_power/KITTY_POWER*100,2)
```

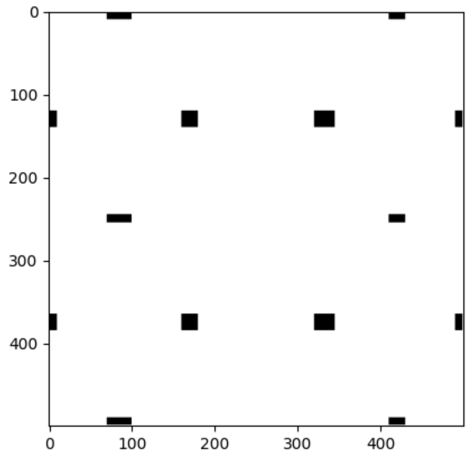
2) There is some periodic noise in “noisy_kitty55.png” image. Remove the periodic noise and calculate PSNR.

Show how to restore the image from periodic noise and display the result,



From reconstructed image, the cat's details are better.

Use notch reject filter to remove periodic noise. So, my filter will reject 14 rectangles. The filter looks like the below fig..



What is the PSNR before adding noise (compared with "kitty55.png")?

```
10.48016049465457
```

PSNR of the restored image

```
8.538423956221498
```

3. Write a paragraph (at least 300 words in Thai or English) to summarize three image applications using frequency analysis in your own words and add the references (excluded

from the total word count). You will have to present and share the applications you found on Wednesday, October 3, 2023 in a discussion session.

answer

Watermarking image in frequency domain. Ordinary technique of watermarking images is in the spatial domain but it’s too obvious to detect by human eye. First step, we will convert the image into freq. domain then we embed the key or watermarks such as the company name in the ASCII format, etc. into the first n pixels of freq. domain then convert it back to spatial domain. The benefit of this type of watermarking is that nobody can detect watermarks with their own eye (the main point is being able to check the ownership of the image because we can encrypt those watermarks in the image and ask user for the key).

Hybrid image. a hybrid image is the image that combines from 2 images which are high-freq. and low-freq. version. if you look at the hybrid image close enough, you will notice the high-frequency image easier. On the other hand, you will notice only low-frequency image if you look at image far away, like 6 feet. In my opinion, It can use to recommend people to get far enough to watch the TV so they can reduce the eye pain from watching too close.

Drawing machine. Let’s explain the problem first. We have a big canvas that includes only a continuous line (drawing that pen always touches the canvas) and only a size of line (can’t push harder to get a thinner line). The “intro to digital image processing” way to store the drawing is stored in the form of 2 Fourier transforms(FT). We can extract the path of the line into 2 continuous movements on the X-axis and Y-axis. We can create signals of X or Y position and time then apply FT to each signal. After we get the FT of each axis movement. We can use those FTs to reconstruct the original image or use them to create an animation of the drawing image.

reference.

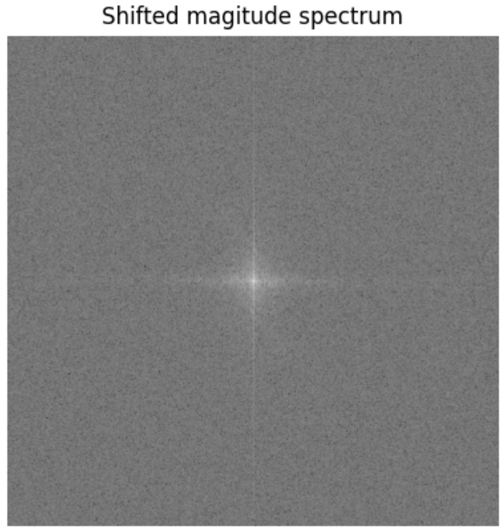
[watermarking in frequency domain]

[hybrid image]

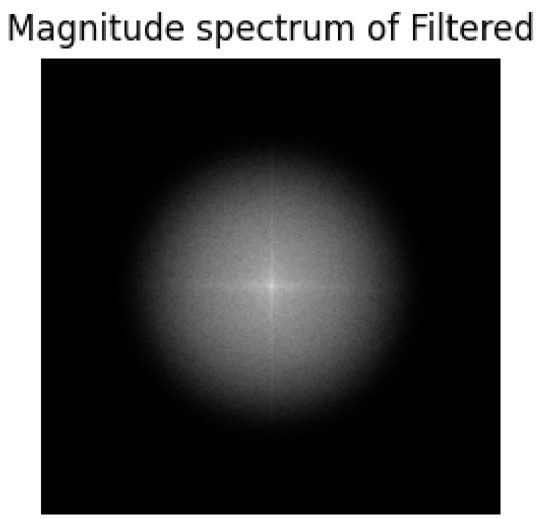
drawing machine

4. Apply Gaussian low-pass filter on a noisy image, “`hillbefore_noise10%.jpg`” below using cutoff frequency, D0 set to 35.

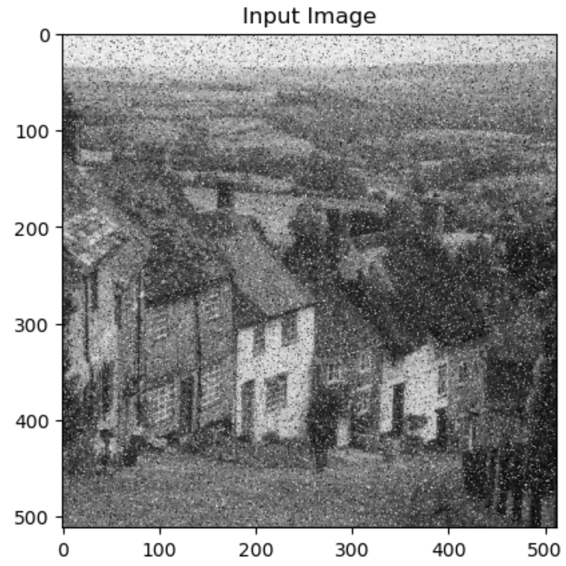
4.1. Generate the Shifted Magnitude Spectrum and put resulted image into the blank box below.

	
Input Image	Shifted Magnitude Spectrum

4.2. Display the Filtered Fourier Spectrum and put the resulted image below.


Filtered Fourier Spectrum

4.3. Show the Image after denoising with the Gaussian low-pass filter method and put the result in the blank box on the right side below.

	
Input Image	Denoised Image

4.4. Discuss the usage of the Gaussian low-pass filter in comparison to other methods, such as Median Blur and Linear Smoothing Filter. Explain the differences and identify which one might be superior. (Short description)

Ans – The type of noise in the image might be S&P noise so the method that is used to denoise should be **Median blur**. So, I will show the image that uses Median blur.

From the image above, The image has more details compared with the image that uses Gaussian blur.