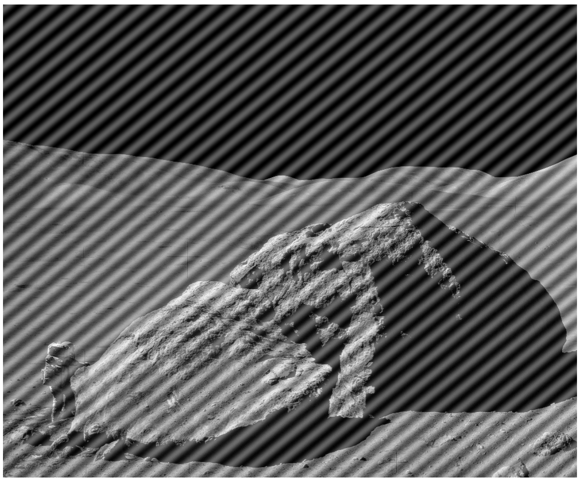
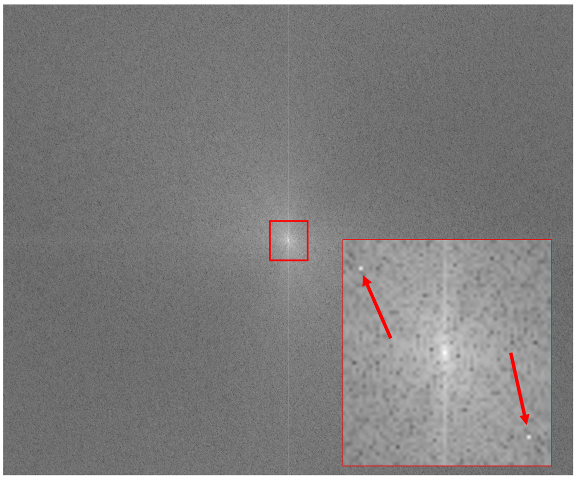
DIP HW 4-2

311505011 人工智慧碩二 黃丰楷

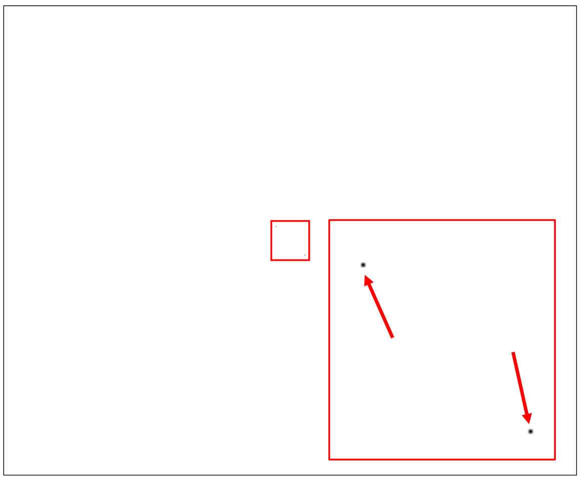
1. Please use FFT and design a frequency filter to cancel the sinusoidal noise of the assigned image, ‘astronaut interference.tif’, and print out the source code and the processed image? (40)



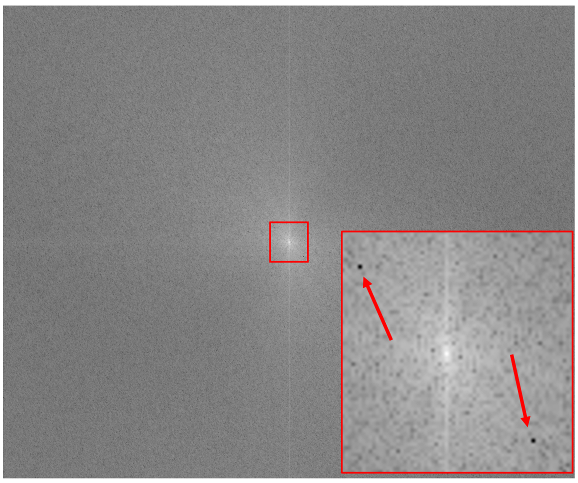
(a) Original image



(b) FFT + Shift



(c) Frequency filter



(d) (b) x (c)

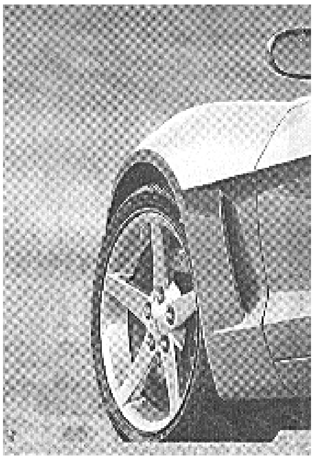
(e) Modified image

- code

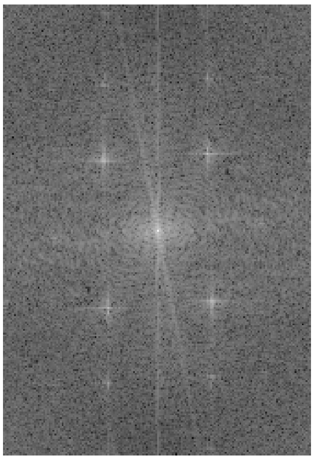
```
import cv2
import numpy as np
import matplotlib.pyplot as plt


def find_max_idx(array, num):
    a = array.copy()
    c = np.zeros(num)

    for i in range(num):
        c[i] = np.max(a)
        if i > 0:
            c[i] = np.max(np.max(a[a<c[i-1]]))
        if i == 0:
            b = np.stack(np.where(a==c[i]),axis=1)
        else:
            b = np.vstack((b,np.stack(np.where(a==c[i]),axis=1)))
    return b

img = cv2.imread("./astronaut-interference.tif", 0)
w, h = img.shape
print(img.shape)
plt.imsave('4_2_1_ori.png', img, cmap='gray')

fft = np.fft.fft2(img)
fft = np.fft.fftshift(fft)
res = np.log(np.abs(fft))
plt.imsave('4_2_1_fftshift.png', res, cmap='gray')
max_idx = find_max_idx(res, 3)
index = max_idx[1:3]
print(index)

mask = np.ones_like(img) / 1.
mask[index[:,0],index[:,1]] = 0.
print(np.where(mask==0))
plt.imsave('4_2_1_mask.png',mask, cmap='gray')
```

```
new_fft = fft * mask
plt.imsave('4_2_1_modfft.png', np.log(np.abs(fft))*mask, cmap='gray')
plt.show()

ifft = np.fft.ifftshift(new_fft)
ifft = np.fft.ifft2(ifft)
plt.imsave( '4_2_1_modimg.png',np.abs(ifft), cmap='gray')
```

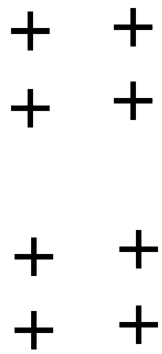
2. Please use FFT and design a frequency filter to cancel the moire-pattern noise of the assigned image, 'car-moire-pattern.tif', and print out the source code and the processed image? (40)



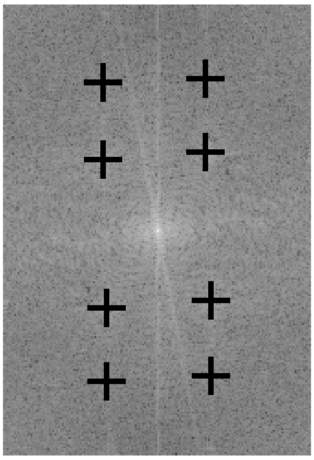
(a) Original image



(b) FFT + Shift



(c) Frequency filter



(d) (b) x (c)

(e) Modified image

- code

```
import cv2
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import numpy as np



# matplotlib read image
img = cv2.imread("car-moire-pattern.tif",0)
print(img.shape)
# Output Images
plt.imsave('4_2_2_ori.png',img, cmap=cm.gray)


# Fourier Transform
fft = np.fft.fftshift(np.fft.fft2(img))
plt.imsave('4_2_2_fftshift.png',np.log(abs(fft) + 1), cmap='gray'), plt.title("fft_img")


mask = np.zeros_like(img)
# left column

mask = cv2.rectangle(mask, (44,41),(64,43), (255,255,255), -1)
mask = cv2.rectangle(mask, (53,32),(55,52), (255,255,255), -1)

mask = cv2.rectangle(mask, (44,83),(64,85), (255,255,255), -1)
mask = cv2.rectangle(mask, (53,74),(55,94), (255,255,255), -1)

mask = cv2.rectangle(mask, (46,164),(66,166), (255,255,255), -1)
mask = cv2.rectangle(mask, (55,155),(57,175), (255,255,255), -1)

mask = cv2.rectangle(mask, (46,204),(66,206), (255,255,255), -1)
mask = cv2.rectangle(mask, (55,195),(57,215), (255,255,255), -1)


# right column
```

```
mask = cv2.rectangle(mask, (100,39),(120,41), (255,255,255), -1)
mask = cv2.rectangle(mask, (109,30),(111,50), (255,255,255), -1)

mask = cv2.rectangle(mask, (100,79),(120,81), (255,255,255), -1)
mask = cv2.rectangle(mask, (109,70),(111,90), (255,255,255), -1)

mask = cv2.rectangle(mask, (103,160),(123,162), (255,255,255), -1)
mask = cv2.rectangle(mask, (112,151),(114,171), (255,255,255), -1)

mask = cv2.rectangle(mask, (103,201),(123,203), (255,255,255), -1)
mask = cv2.rectangle(mask, (112,192),(114,212), (255,255,255), -1)


mask = 255 - mask
mask = mask/np.max(mask)
plt.imsave('4_2_2_mask.png',mask, cmap='gray'), plt.title("mask")




new_fft = fft*mask
plt.imsave('4_2_2_modfft.png',np.log(abs(new_fft) + 1), cmap='gray'), plt.title("modify fft_img")



## inverse Fourier Transform
new_img = np.fft.ifft2(np.fft.ifftshift(new_fft))
plt.imsave('4_2_2_modimg.png',np.abs(new_img), cmap='gray'), plt.title("ifft_img")
```

3. Please comment and compare your two design freq. filters? (20)

In Q1, I firstly applied FFT conversion to transform the Fig. (a) into the frequency domain. Subsequently, I conducted a Frequency shift to reposition the low-frequency response at the center for filter design, resulting in Fig. (b). Upon examination, I noticed two highly responsive pixels located at the top left and bottom right corners of the center in Fig. (b). I conducted a maximum search and confirmed that these two pixels were the next most responsive. Following this, I designed a frequency filter (Fig. (c)) with a value of 0 at these two pixels and 1 at all other locations. After designing the filter, I multiplied it with the original FFT to effectively filter out noise, resulting in Fig. (d). Finally, I performed inverse displacement and Inverse FFT to bring the image back to the spatial domain, ultimately yielding a clean image depicted in Fig. (e).

In Q2, initially, I conducted an FFT conversion to transform image (a) into the frequency domain. Subsequently, I applied Frequency shift to reposition the low-frequency response at the center for filter design, resulting in image (b). Upon closer examination, I identified four vertical noise distributions on the left and right sides of the center of Fig. (b), forming a cross-shaped pattern. To address this, I designed a frequency filter (Fig. (c)) and created cross-shaped masks to target these cross-shaped noises. The value on these crosses was set to 0, while all other positions were set to 1. After designing the filter, I multiplied it with the original FFT to effectively filter out the noise (Figure (d)). Finally, I performed inverse displacement and Inverse FFT to bring the image back to the spatial domain, ultimately yielding a clean image depicted in Fig. (e).

The noise patterns in the Q1 and Q2 images exhibit distinct characteristics - diagonal distribution and lattice-like noise, respectively. Dealing with these using a spatial filter would prove challenging, but employing a frequency filter offers a more straightforward solution. Additionally, the frequency response of these two images varies significantly. In the case of Q1, the presence of periodic sine noise manifests as two simple dots in the frequency domain. Conversely, the noise in Q2 is more intricate, appearing irregular in the frequency domain. Nevertheless, it can still be discerned that it assumes a cross-shaped pattern, aligning with the lattice-like noise observed in the spatial domain.