Digital Image Processing (2021 spring)
HOMEWORK ASSIGNMENT #4 Digital Halftoning, Frequency Domain

environment: C++, python3, opencv4

Law Po Ying b06902091

Problem 1: DIGITAL HALFTONING

(a) Perform dithering using the dither matrix I2 in Figure 1.(b) and output the result as result1.png.

Determine the threshold matrix T2 from dither matrix I2 using the equation below and apply it to the image.

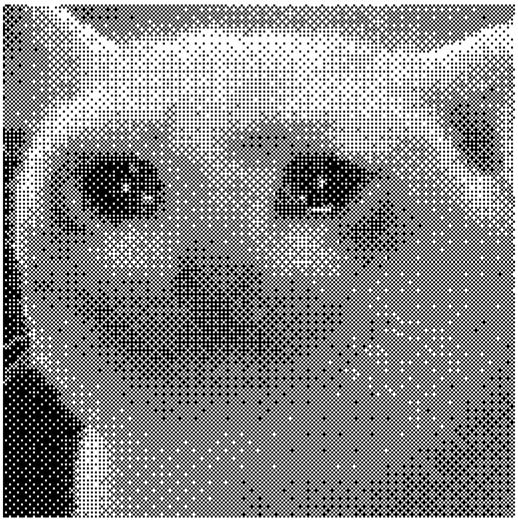
$$T(i,j) = 255 \cdot \frac{I(i,j)+0.5}{N^2}$$

result1.png

(b) Expand the dither matrix I2 to I256 (256 × 256) and use it to perform dithering. Output the result as result2.png. Compare result1.png and result2.png along with some discussions.

Expand the dither matrix I2 into I256 using the general form below. Then, determine its threshold matrix T256 and apply it to the image.

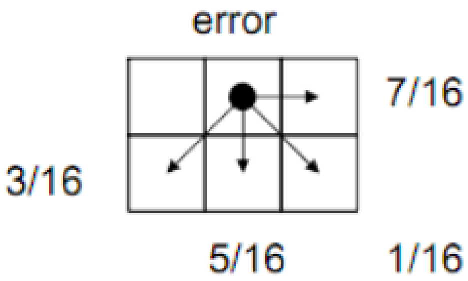
$$I_{2n}(i,j) = \begin{bmatrix} 4I_n(i,j)+1 & 4I_n(i,j)+2 \\ 4I_n(i,j)+3 & 4I_n(i,j)+0 \end{bmatrix}$$



result2.png

We can observe that the halftoned image using a larger size of dither matrix(like result2.png) is much more detailed and shows more levels of intensity. Alternatively, the intensity performance of the halftoned image using a smaller size of dither matrix is relatively weaker.

(c) Perform error diffusion with two different filter masks. Output the results as result3.png, and result4.png, respectively. Discuss these two masks based on the results. You can find some masks here (from lecture slide 06. p23)

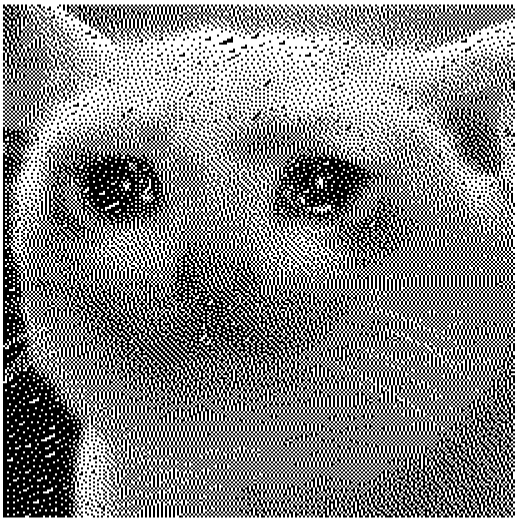


Floyd Steinberg filter mask

error

		●	7	5	
3	5	7	5	3	
1	3	5	3	1	(1/48)

Jarvis et al filter mask



result3.png

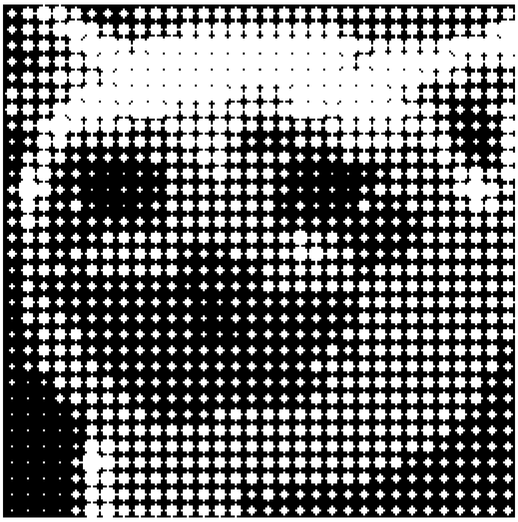
result4.png

For Jarvis's filter mask, we can observe that the dots are much averaged distributed in the halftoned image. And for Floyd Steinberg's filter mask, we can observe that there are some dots linked up together and cause some black "items", which is relatively unsightly. And I think the resultant image of Jarvis filter mask has a smoother texture.

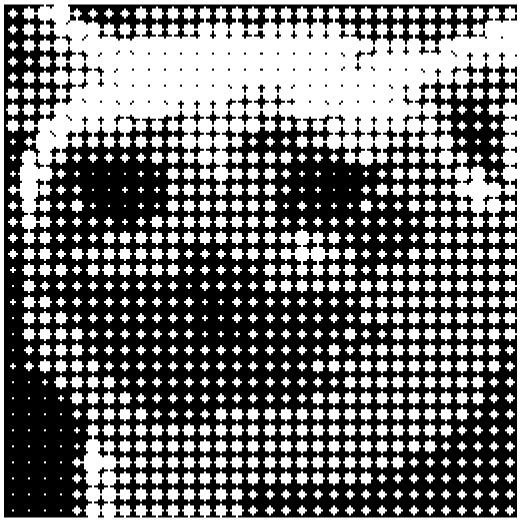
(d) Try to transfer result1.png to a dotted halftone / manga style binary image such as sample1 dotted.png in Figure 1.(c). Describe the steps in detail and show the result. You may need to utilize a function like cv2.circle to draw a circle.

<u>Steps</u>

1. Takes 8x8 pixels as a group. For each group, determine its mean or median of grayscale values.
2. Normalize the mean/median into [0,6]. (generally [0,4~6])
3. Let the normalized mean/median be the radius of the dot and draw the dot on the center of the 8x8 pixel group.



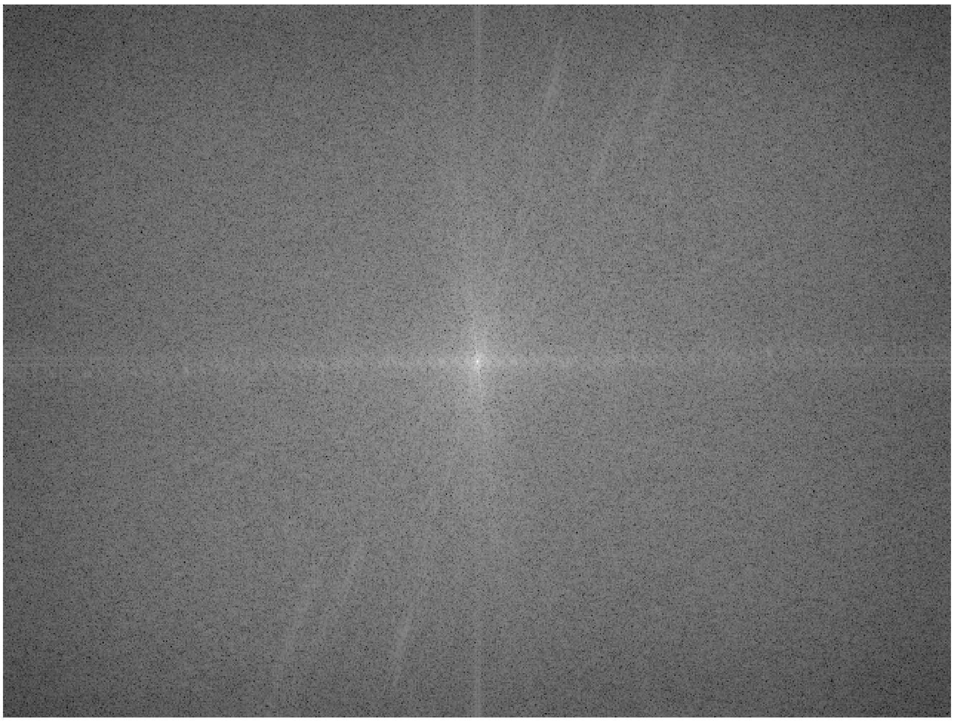
sample1 dotted_mean.png



sample1 dotted_median.png

Problem 2: FREQUENCY DOMAIN

(a) **Perform Fourier transform on sample2.png to obtain its frequency spectrum and output it as result5.png. (Please take the log magnitude of the absolute value and center the low frequency part at the origin for visualization.)**



result5.png

(b) Based on the result of part (a), design and apply a low-pass filter in the frequency domain and transform the result back to the pixel domain by inverse Fourier transform. The resultant image is saved as result6.png. Please also design a low-pass filter in the pixel domain which behaves similarly to the one you design in the frequency domain. Output the result as result7.png and provide some discussions.

Using the Ideal low-pass filtering method to implement low-pass filtering in the frequency domain.

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

$$D(u, v) = \left[(u - M/2)^2 + (v - N/2)^2\right]^{1/2}$$

low-pass filtering in frequency domain with radius=30

result6.png (radius=65)

result7.png (low-pass filtering)

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \frac{1}{25}$$

mask

The resultant image has some grid-like pattern when applying a low-pass filter in the frequency domain. However, the resultant image of low pass filtering in the pixel domain is much smoother and the contour is linked up.

(c) Based on the result of part (a), design and apply a high-pass filter in the frequency domain and transform the result back to the pixel domain by inverse Fourier transform. The resultant image is saved as result8.png. Please also design a high-pass filter in the pixel domain which behaves similarly to the one you design in the frequency domain. Output the result as result9.png and provide some discussions.

Using the Ideal high-pass filtering method to implement high-pass filtering in the frequency domain.

$$H(u,v)=\frac{1}{1+[D(u,v)/D_0]^{2n}}$$

$$D(u,v)=\left[(u-M/2)^2+(v-N/2)^2\right]^{1/2}$$

high pass filtering in frequency domain with Cutoff=15

result8.png (Cutoff=80)

result9.png (high-pass filtering)

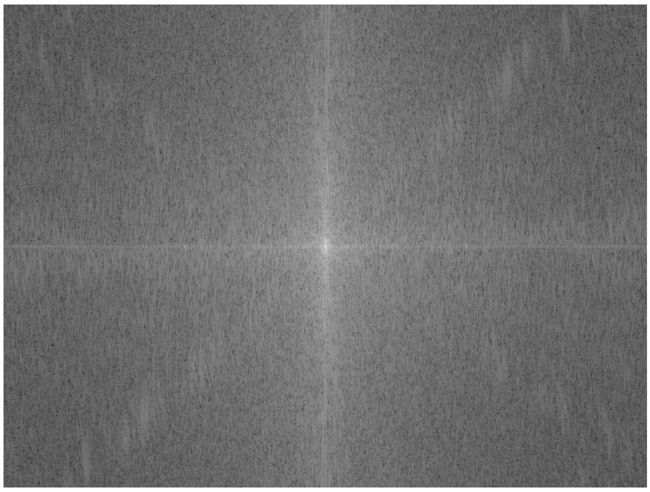
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

mask

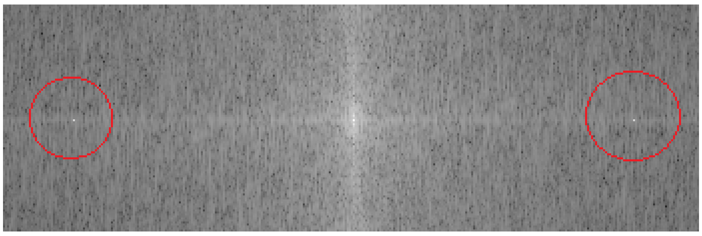
The resultant image, which uses high-pass filtering in the pixel domain, has a sharper and clearer contour. The edges are well-connected. However, the resultant image in the frequency domain has a relatively blurry contour.

(d) Perform Fourier Transform on sample3.png and output it as result10.png. Please discuss what you observe in sample3.png and result10.png.

sample3.png



result10.png



We can see some undesired vertical lines in sample3.png. After Fourier transforms, we can see that those vertical lines presented two dots {(180,240), (460,240) }in the frequency spectrum.

(e) Try to remove the undesired pattern on sample3.png and output it as result11.png.

<u>Steps</u>

1. Perform Fourier Transform on sample3.png
2. Erase the two dots in the frequency spectrum of sample3.png
3. Transform the frequency spectrum back to the pixel domain by inverse Fourier transform.

result11.png

Reference / supplement:

1. https://numpy.org/doc/stable/reference/generated/numpy.fft.fft2.html
2. https://blog.csdn.net/amusi1994/article/details/79529870
3. https://stackoverflow.com/questions/38025838/normalizing-images-in-opencv
4. https://www.cs.unm.edu/~brayer/vision/fourier.html
5. https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm
6. https://www.projectrhea.org/rhea/index.php/The_2-D_Fourier_Transform_and_Images