گروه كامپيوتر - دانشكده مهندسي - دانشگاه فردوسي مشهد

مستندات گزارش تمرین چهارم

درس مبانی بینایی کامپیوتر

**ارائه ‌دهنده:**

**صالح شیروانی**

**استاد درس:**

**دکتر امیرحسین طاهری نیا**

**پاییز1401**

چکیده

در این تمرین می‌خواهیم تصاویر را در حوزه فرکانس بررسی کنیم. کلیدی ترین ابزار که کمک زیادی در این مسیر به خواهد کرد تبدیل فوریه است که در ادامه به تفضیل از آن استفاده خواهیم کرد و مفاهیمی که تاکنون به صورت تئوری آموخته ایم را در عمل به کار خواهیم برد. در بخشی از تمرین به بررسی ضرایب فوریه خواهیم پرداخت و خواهیم دید که این ضرایب چه اطلاعات مفیدی را در اختیار ما قرار می‌دهند. در نهایت در قسمت دوم تمرین، فیلترینگ در حوزه ی فرکانس را بررسی خواهیم کرد و خواهیم دید که دستکاری ضرایب فوریه می‌تواند چه تاثیراتی روی تصویر داشته باشد.

Technical Description

**4- Frequency Domain**

4-1- Fourier Transform

*4-1-1- در این سوال برای محاسبه ی تبدیل فوریه از تابع dft موجود در cv2 استفاده می‌کنیم. همچنین برای به دست آوردن magnitude می‌توان از تابع magnitude موجود در cv2 بهره برد. همچنین برای شیفت دادن فرکانس dc از تابع fftshift موجود در fft که به ماژول numpy تعلق دارد استفاده می‌کنیم.*

*4-1-2- در این تمرین قصد داریم تبدیل فوریه 2 بعدی روی تصویر اعمال کنیم. برای این کار از تابع dft موجود در cv2 استفاده می‌کنیم. همچنین برای به دست آوردن magnitude می‌توان از تابع magnitude موجود در cv2 بهره برد. در این تمرین همچنین از ما خواسته شده است تاثیر شیفت دادن فرکانس dc را بررسی کنیم که برای این کار از تابع fftshift موجود در fft که به ماژول numpy تعلق دارد استفاده می‌کنیم. در نهایت تاثیر اعمال تبدیل لگاریتم را روی تصویر magnitude بررسی خواهیم کرد. در قسمت نتایج گزارش (Results) که در ادامه آمده است درباره ی نتایج به دست آمده از این تمرین بیشتر بحث خواهیم کرد.*

*4-2-* Filtering

*4-2-1- در قسمت اول تمرین که خواسته شده سایز مورد نیاز DFT برای محاسبه کانولوشن یک تصویر و یک فیلتر را محاسبه کنیم.*

*سایز مورد نیاز DFT برای به دست آوردن کانولوشن یک تصویر و یک فیلتر حداقل در هر* ***بعد*** *است. این اندازه DFT برای اطمینان از اینکه نتیجه کانولوشن حاوی تمام اطلاعات است، کافی است. البته اندازه رایج برای DFT معمولاً توان بعدی 2 است که در این سوال 512 خواهد بود.*

مراحل بدست آوردن نتیجه کانولوشن با استفاده از DFT به شرح زیر است:

* اندازه تصویر و فیلتر را به اندازه ای برابر یا بیشتر از اندازه DFT مورد نیاز (در این سوال حداقل) تبدیل کنید. (اعمال padding)
* *DFT دوبعدی تصویر و فیلتر پد شده را محاسبه کنید.*
* *DFT تصویر و DFT فیلتر را ضرب کنید تا حاصل ضرب نقطه ای آنها را بدست آورید. ( ضرب آرایه ای)*
* *برای بدست آوردن نتیجه کانولوشن، DFT دو بعدی معکوس حاصل ضرب نقطه ای را محاسبه کنید.*
* *قسمت های اضافی (ناشی از padding) را حذف کنید و نتیجه را استخراج کنید.*

*برای اینکه برابر باشد، باید مقادیر در بازه ی باشند. این به این دلیل است که ما با DFT یک سیگنال گسسته با دوره تناوب 256 سروکار داریم و نتیجه کانولوشن فقط در اولین دوره تناوب به طور کامل نشان داده می شود.*

**Results**

تابع زیر تابع مهمی است که در این تمرین از آن زیاد استفاده شده است

def DFT(*image*):

*# Pad the image with zeros*

    img\_padded = np.pad(*image*, [(0, max(*image*.shape) - i) for i in *image*.shape], *mode*='constant')

*# Convert the image to a float32 type*

    img\_padded = np.float32(img\_padded)

*# Compute the 2D DFT*

    dft = cv2.dft(img\_padded, *flags*=cv2.DFT\_COMPLEX\_OUTPUT)

    return dft

**4- Frequency Domain**

4-1- Fourier Transform

4-1-1-

def ft\_filter(*filter*):

*# Pad the filter to the size of the image, if necessary*

*filter* = cv2.copyMakeBorder(*filter*, 0, 0, 0, 0, cv2.BORDER\_CONSTANT, *value*=0)

*# Apply the 2D DFT to the filter*

    dft = cv2.dft(np.float32(*filter*), *flags*=cv2.DFT\_COMPLEX\_OUTPUT)

*# Shift the zero-frequency component to the center of the spectrum*

    dft\_shift = np.fft.fftshift(dft)

*# Compute the magnitude spectrum of the DFT*

    magnitude\_spectrum = 20 \* np.log(cv2.magnitude(dft\_shift[:,:,0], dft\_shift[:,:,1]))

    return magnitude\_spectrum

filter1 = np.array([

    [1, 2, 1],

    [2, 4, 2],

    [1, 2, 1]

])/16

filter2 = np.array([

    [-1, -1, -1],

    [-1,  8, -1],

    [-1, -1, -1]

])

filter3 = np.array([

    [0, -1,  0],

    [-1, 5, -1],

    [0, -1,  0]

])

filter1\_dft = ft\_filter(filter1)

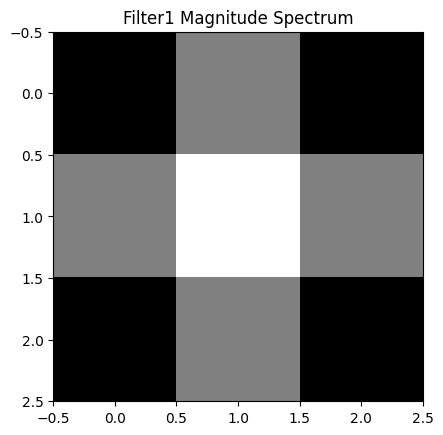
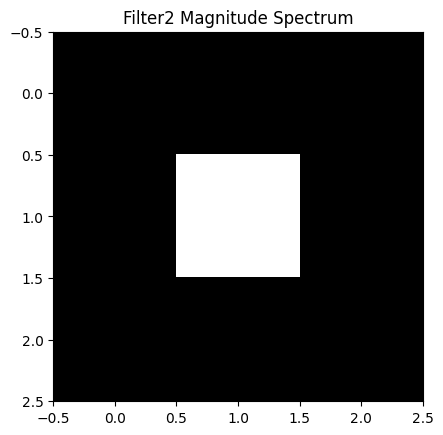
filter2\_dft = ft\_filter(filter2)

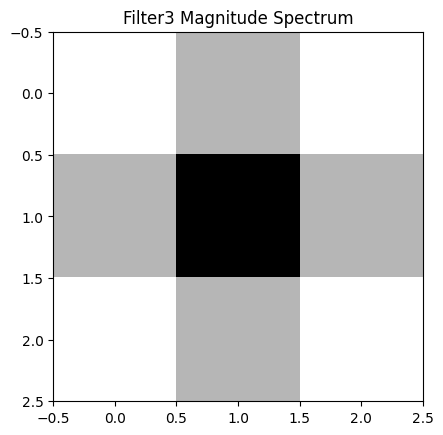
filter3\_dft = ft\_filter(filter3)

plot\_image(filter1\_dft, 'Filter1 Magnitude Spectrum')

plot\_image(filter2\_dft, 'Filter2 Magnitude Spectrum')

plot\_image(filter3\_dft, 'Filter3 Magnitude Spectrum')

تصاویر خروجی به صورت زیر هستند:



*4-1-2-*

def DFT\_log\_shift(*image*):

    dft = DFT(*image*)

*# Compute the magnitude of the DFT coefficients*

    magnitude\_spectrum = cv2.magnitude(dft[:, :, 0], dft[:, :, 1])

*# Shift the zero-frequency component to the center of the image*

    dft\_shift = np.fft.fftshift(dft)

*# Compute the magnitude of the DFT coefficients*

    magnitude\_spectrum\_shifted = cv2.magnitude(

        dft\_shift[:, :, 0], dft\_shift[:, :, 1])

    return magnitude\_spectrum, magnitude\_spectrum\_shifted

def plot\_result(*magnitude\_spectrum*, *magnitude\_spectrum\_shifted*):

*# Plot the magnitude of the DFT coefficients with shifting and without logarithmic transformation*

    plt.figure(*figsize*=(10, 5))

    plt.subplot(121), plt.imshow(*magnitude\_spectrum\_shifted*, *cmap*='gray')

    plt.title('Magnitude Spectrum (Shifted)'), plt.xticks([]), plt.yticks([])

*# Plot the magnitude of the DFT coefficients without shifting and without logarithmic transformation*

    plt.subplot(122), plt.imshow(*magnitude\_spectrum*, *cmap*='gray')

    plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])

    plt.show()

*# Plot the magnitude of the DFT coefficients with shifting and with logarithmic transformation*

*magnitude\_spectrum\_shifted* = np.log(*magnitude\_spectrum\_shifted*)

    plt.figure(*figsize*=(10, 5))

    plt.subplot(121), plt.imshow(*magnitude\_spectrum\_shifted*, *cmap*='gray')

    plt.title('Magnitude Spectrum (Shifted & Logarithmic)'), plt.xticks(

        []), plt.yticks([])

*# Plot the magnitude of the DFT coefficients without shifting and with logarithmic transformation*

*magnitude\_spectrum* = np.log(*magnitude\_spectrum*)

    plt.subplot(122), plt.imshow(*magnitude\_spectrum*, *cmap*='gray')

    plt.title('Magnitude Spectrum (Logarithmic)'), plt.xticks(

        []), plt.yticks([])

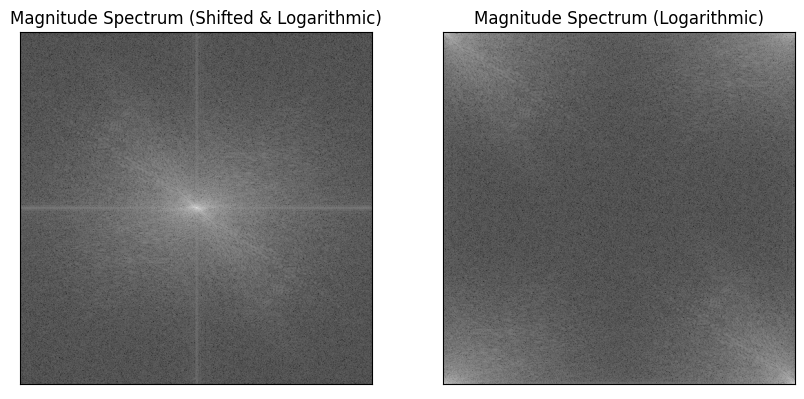
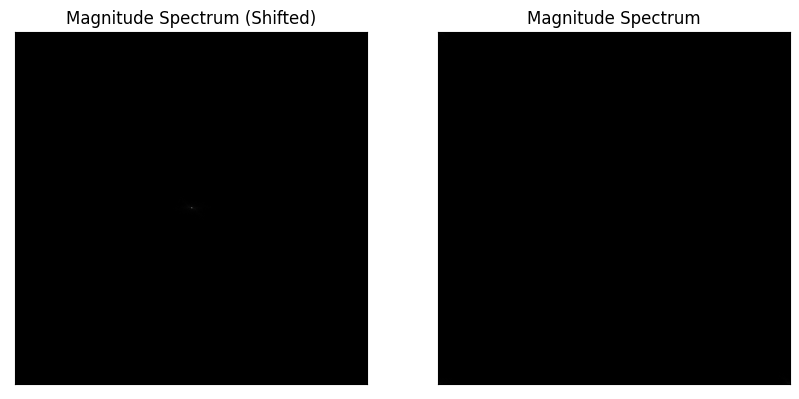
    plt.show()

magnitude\_spectrum, magnitude\_spectrum\_shifted = DFT\_log\_shift(lena)

plot\_result(magnitude\_spectrum, magnitude\_spectrum\_shifted)

*تصاویر خروجی مربوط به این تمرین به صورت زیر است:*

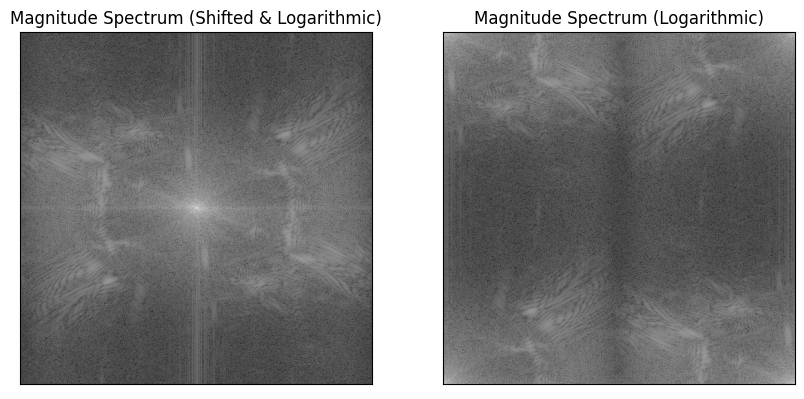
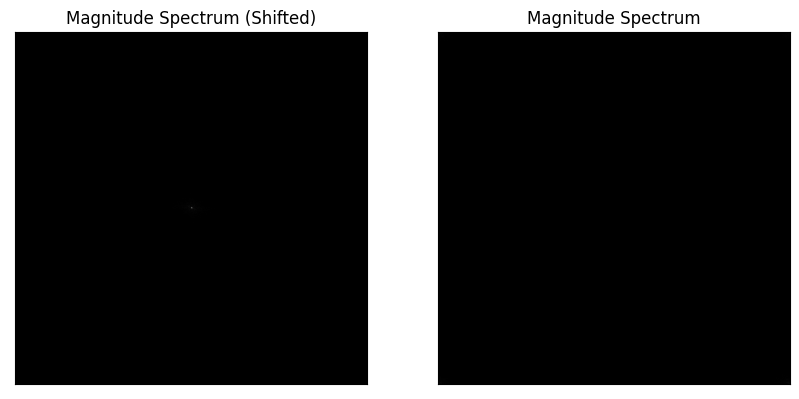
***Lena***

**

magnitude\_spectrum, magnitude\_spectrum\_shifted = DFT\_log\_shift(Barbara)

plot\_result(magnitude\_spectrum, magnitude\_spectrum\_shifted)

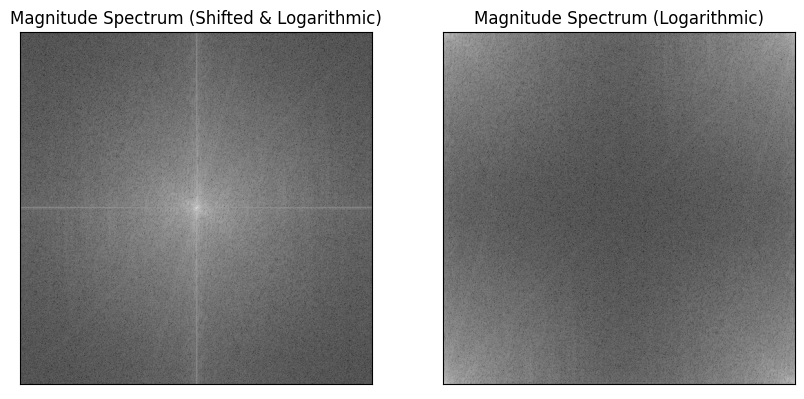
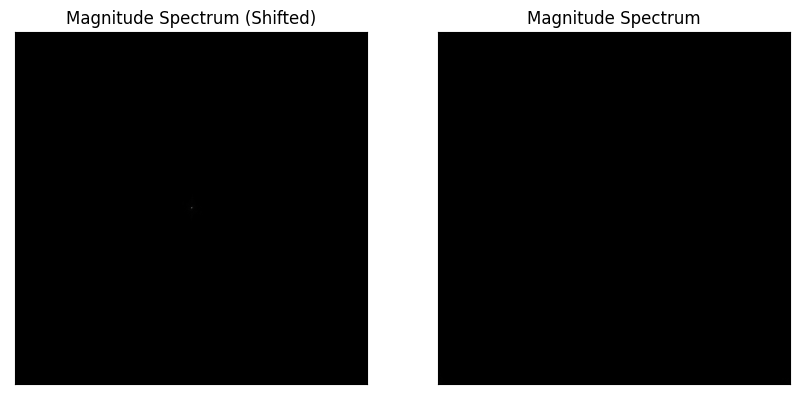
***Barbara***

**

magnitude\_spectrum, magnitude\_spectrum\_shifted = DFT\_log\_shift(F16)

plot\_result(magnitude\_spectrum, magnitude\_spectrum\_shifted)

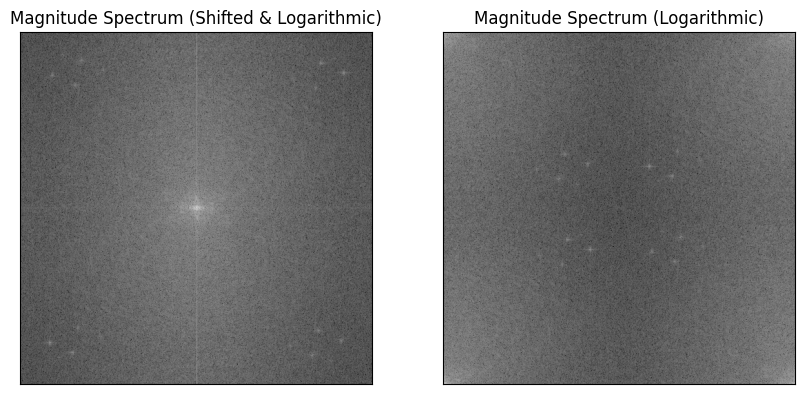
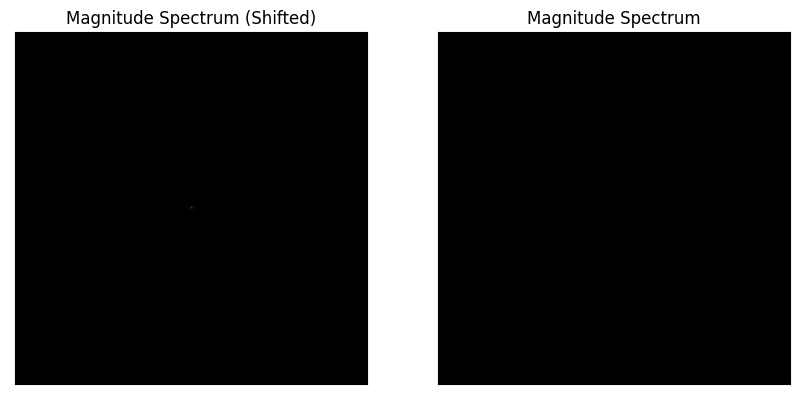
***F16***

******

magnitude\_spectrum, magnitude\_spectrum\_shifted = DFT\_log\_shift(Baboon)

plot\_result(magnitude\_spectrum, magnitude\_spectrum\_shifted)

***Baboon***

******

*در حوزه فرکانس، ضرایب DFT با بزرگی های زیاد با فرکانس های غالب موجود در تصویر مطابقت دارد. فرکانس‌های غالب می‌توانند فرکانس‌های پایین (لبه‌ها و بافت‌ها)، فرکانس‌های بالا (جزئیات و نویز) یا ترکیبی از هر دو باشند.*

*به عنوان مثال، اگر یک تصویر حاوی لبه های زیادی باشد، به احتمال زیاد فرکانس های پایین دارای بزرگی های زیادی خواهند بود، زیرا لبه ها با تغییرات در مقادیر شدت مشخص می شوند، که می تواند با ضرایب فرکانس پایین در حوزه فرکانس نشان داده شود. از طرف دیگر، اگر یک تصویر حاوی نویز زیادی باشد، فرکانس‌های بالا دارای مقادیر زیادی خواهند بود، زیرا نویز با نوسانات سریع و تصادفی در مقادیر شدت مشخص می‌شود که می‌تواند با ضرایب فرکانس بالا در حوزه فرکانس نشان داده شود.*

*4-2-* Filtering

*4-2-1- توضیح کامل مربوط به این قسمت در شرح فنی (technical description) مربوط به این سوال در همین گزارش ارائه شد.*

***پیوست***

* *کدها و تصاویر مربوط به این تمرین در*[*GitHub*](https://github.com/saleh-sh/HW4) *در دسترس است.*