تمرین ۴ بینایی - فوریه

سارا قوام پور

اطلاعات گزارش

تاریخ:

واژگان کلیدی:
تبدیل فوریه
حوزه فوریه
فیلترینگ در حوزه فرکانس
فیلتر بالا گذر
نوبز
لبه
معکوس تبدیل

چکیده

در این تمرین به واسطه سؤال‌ها با تبدیل فوریه و فیلترینگ در حوزه فرکانس آشنا میشویم. در حوزه فوریه میتوان عملیات هایی مانند noise detection, noise reduction را ساده‌تر انجام داد. همچنین فیلتر های بهبود لبه یا پایین گذر هم راحت‌تر اعمال میشوند.

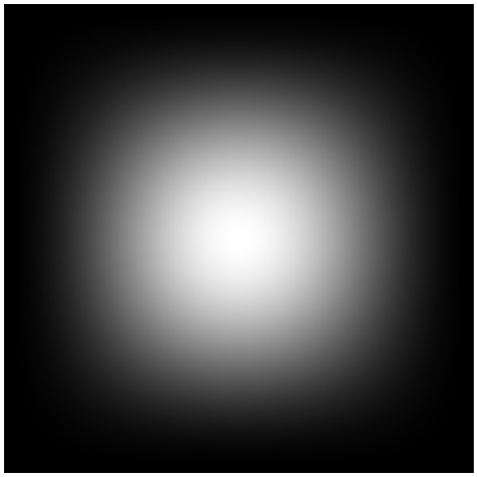
۱-مقدمه

این داکیومنت شامل پیاده‌سازی تمارین سری ۴ قوریه با استفاده از زبان پایتون میباشد. برای هر تمرین نتایج به صورت عکس با جدول ارائه شده‌اند و بر روی نتایج تحلیل صورت گرفته است.

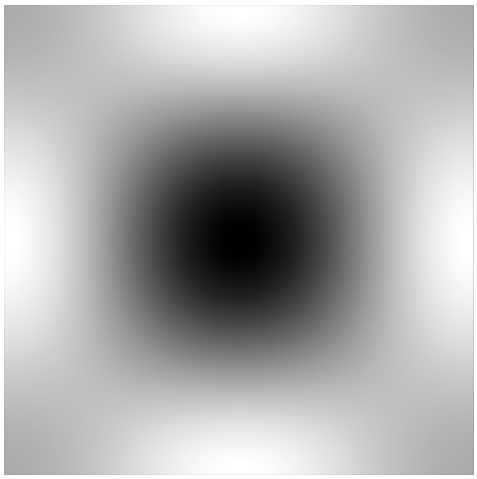
۲-بررسی فنی تمرینات و نتایج

۲-۱ تمرین ۱-۱-۴

در این تمرین ۳ فیلتر داده شده است که باید ابتدا طیف آن‌ها و عمل‌کرد آن‌ها و اینکه جداپذیر هستند بحث شود و سپس بر روی تصویر در حوزه فرکانس اعمال شوند.
مرحله اول نمایش طیف این فیلتر ها میباشد.
برای نمایش طیف فیلتر ها تابع fourier_magnitude نوشته شده است که تبدیل فوریه گسسته DFT را بر روی فیلتر اعمال میکند با استفاده از np.fft.fft2. سپس برای بهتر دیده شدن با استفاده از fftshift پس از تبدیل فوریه مرکز فرکانسی را به مرکز مختصاتی شیفت میدهد.سپس برای نشان دادن طیف باید مقدار magnitude را با استفاده از np.abs به دست آورد و نمایش داد. خروجی مربوط به طیف ۳ فیلتر :

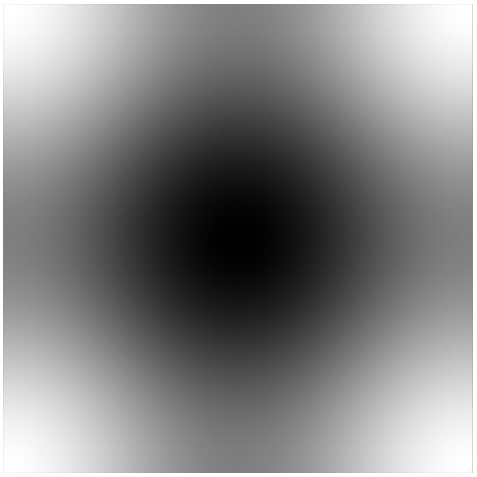


شکل ۱)طیف فیلتر a



شکل۲)طیف فیلتر b

* [1] ghavams2001@gmail.com



شکل ۳)طیف فیلتر c

با توجه به شکل ۱ مشاهده می‌شود که این فیلتر یک فیلتر پایین گذار است چون در نواحی مربوط به فرکانس های پایین در مرکز مقادیر بیشتری دارد و در نواحی مربوط به فرکانس بالا مقدار کمی دارد. به این معنا که لبه ها و نویز ها را که فرمانس بالا دارند را تضعیف میکنند وفرکانس های پایین را عبور میدهد.

همچنین برای فیلتر a از روی مقادیر ضرایب نیز میتوان تشخیص داد که این فیلتر یک فیلتر میانگین گوسی است که به پیکسل مرکزی وزن بیشتری میدهد و با دور شدن از مرکز ضرایب کاهش میابند. فیلتر a از نوع smoothing است

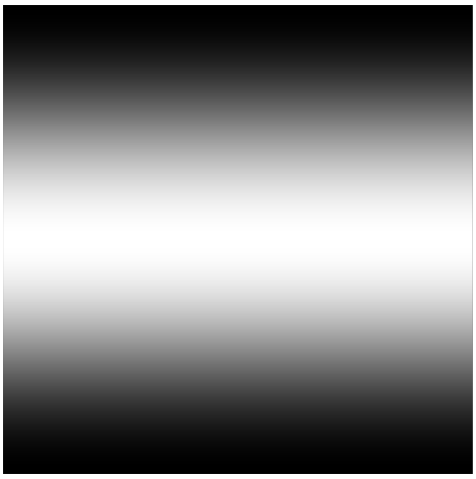
با توجه به شکل ۲ مشاهده می‌شود که این فیلتر یک فیلتر بالا گذار است چون مقادیر مرکزی که فرکانس پایینی دارند را به صورت کامل عبور نمیدهد و فرکانس ها بالا را عبور میدهد.

از روی ضرایب فیلتر b نیز میتوان متوجه شد که این فیلتر در واقع فیلتر لبه یاب مشتق دوم ۲ بعدی لاپلاسین میباشد که لبه های افقی, عمودی و قطری را میابد و درواقع اختلاف هر پیکسل را با ۸ پیکسل اطرافش به دست می‌آورد و درواقع یک فیلتر edge detector است.

از روی ضرایب فیلتر c نیز میتوان تشخیص داد که این یک فیلتر لاپلاسین افقی و عمودی است که ضریب مرکزی آن یکی بیشتر است. به این معنا که درواقع با اعمال این فیلتر لبه های پیدا شده توسط فیلتر لاپلاسین به صورت همزمان به عکس اضافه می‌شوند و لبه هارا تقویت میکنند و به این دلیل فیلتر c یک فیلتر edge enhancement است.

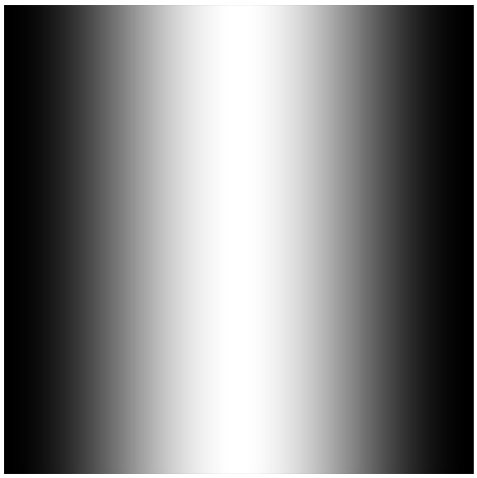
در این قسمت جدا پذیر بودن و جدا پذیر نبودن فیلتر ها را بررسی میکنیم. به طور کلی فیلتری جدا پذیر است که بتوان سطر ها و ستون‌های آن را به عنوان مضربی از هم نوشت. در بین این ۳ فیلتر تنها فیلتر a جدا پذیر است و دو فیلتر دیگر جدا پذیر نیستندو فیلتر a را میتوان به صورت حاصلضرب مانریس ستونی با ضریب ۱/۴(فیلتر a_1) و ماتریس سطری یا ضریب ۱/۴ (فیلتر a_2) تبدیل کرد.

طیف های این ۲ فیلتر به صورت زیر است:



شکل۴) طیف فیلتر a_1 (قسمت ستونی فیلتر a)

مشاهده می‌شود که طیف فیلتر a_1 درواقع لبه های افقی را نشان میدهد و به این معناست که این فیلتر در حوزه مکان لبه های عمودی را نشان میدهد چون عمل‌کرد در حوزه فرکانس و مکان برعکس هم میباشد. به عبارتی این فیلتر در حوزه مکان به صورت عمودی میان گیری میکند.



شکل۵) طیف فیلتر a_2 (قسمت سطری فیلتر a)

شکل ۵ نشان میدهد که طیف فیلتر قسمت سطری فیلتر a (فیلتر a_2) لبه ای عمودی را نشان میدهد و این به این معناست که این فیلتر در حوزه مکان لبه های افقی را نشان میدهد چون عمل‌کرد در حوزه فرکانس و مکان برعکس هم میباشد. به عبارتی این فیلتر در حوزه مکان به صورت افقی میان گیری میکند.
با ترکیب فیلتر a_1 وa_2 فیلتر a یه دست می‌آید که یک فیلتر low pass است.

در این مرحله این فیلتر ها در حوزه فرکانس به عکس اعمال کرده و نتایج را بررسی میکنیم.

برای اعمال کردن فیلتر ها در حوزه فرکانس باید به گونه‌ای عکس را pad کرد که اگز سایز اصلی تصویر m در n است, خروجی padding سایز 2m در 2n داشته باشد و عکس اصلی در بالا سمت چپ قرار داشته باشد. برای padding تابع pad_before_fft نوشته شده است.
این تابع در ورودی عکس یا فیلتر مورد نظر که باید pad شود را دریافت میکند و ورودی دوم آن سایز عکس ورودی است که خروجی باید به دو برابر سایز آن برسد(pad_size) . ابتدا یک آرایه تمام صفر به اندازه ۲ برابر pad_size ساخته می‌شود و عکس ورودی در بالا سمت راست این آرایه قرار میگیرد.

پیاده‌سازی فیلترینگ در حوزه تبدیل فرکانس در تابع filter_frequency_domain انجام شده است. این تابع به عنوان ورودی عکس مورد نظر و فیلتر را در یفت میکند. ابتدا برای پیش پردازش باید با استفاده از تایع pad_before_fft عکس وردی را به شیوه ای که توضیح داده شد pad کنیم.
پس از پدینگ عکس ورودی به عنوان مرحله پیش پردازش برای فیلترینگ در حوزه فرکانس, با استفاده از fft2 در کلاس np.fft تبدیل DFT را روی عکس ورودی اعمال میکنیم و سپس به منظور سیف دادن مرکز فرکانسی به مرکز تصویر و نمایش بهتر بعد از تبدیل از fftshift استفاده می‌شود اکنون تصور در حوزه فرکانس آماده است و نوبت آماده سازی فیلتر است.
فیلتر نیز چون باید در سایز عکس pad شده باشد, توسط تایع pad_before_fft بر روی آن پدینگ صورت میگیرد در حوزه مکان. سپس با استفاده از fft2 فیلتر را به حوزه فرکانس میبریم و با استفاده از fftshift مرکز فرکانسی و مختصاتی و فیلتر را بر هم منطبق میکنیم .
حالا که فیلتر و عکس در حوزه فرکانس آماده هستند میتوان فیلترینگ را انجام داد.

فیلترینگ که در حوزه مکان که با convolve فیلتر در عکس انجام میشد در حوزه فرکانس به راحتی با ضرب آرایه ای فیلتر در حوزه فرکانس با عکس در حوزه فرکانس انجام میشود.
پس از ضرب آرایه ای فیلتر در حوزه فرکانس با عکس در حوزه فرکانس, ابتدا حاصل را با ifftshift شیفت میدهیم تا دوباره مرکز فرکانسی در بالا سمت چپ قرار بگیرد و بعد با استفاده از ifft2 معکوس تبدیل فوریه گسسته را انجام داده تا به حوزه مکان برسیم. چون خروجی ifft2 نیز اعداد مختلط هستد باید magnitude آن را به صورت np.abs به عنوان خروجی از تابع filter_frequency_domain برگردانیم. خروجی درواقع حاصل فیلترینگ است که مراحل آن در حوزه فرکانس انجام شد به جهت راحتی و در انتها با معکوس تبدیل به حوزه مکان برگشت.
چون فیلتر a جدا پذیر است, فیلتر های a_1 و a_2 را به صورت جداگانه به تصویر اعمال میکنیم.
نتایج اعمال فیلترهای این تمرین در حوزه فرکانس:

شکل۶)تصویر اصلی grayscale از Lena

شکل۷)نتیجه اعمال فیلتر a بر روی عکس lena

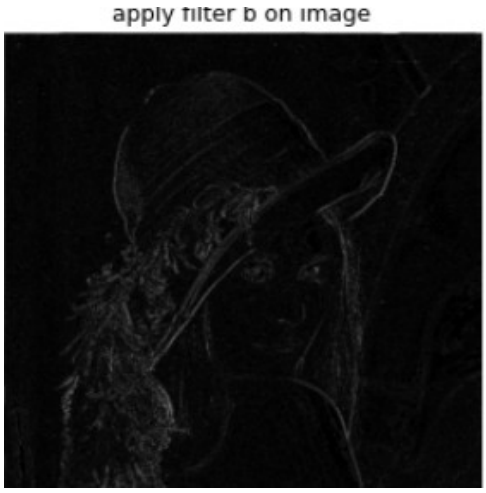
طبق شکل ۷ مشخص می‌شود که در این تصویر لبـه هـا و نویز ها تضعیف شده‌اند و تصویر smooth شده است.

شکل۸)نتیجه اعمال فیلتر a_2

طبق شکل ۴ و ۸ میتوان نتیجه گرفت کـه فیلـتر a-1 در جهت عمودی میانگین گیری کرده است چون مقـداری در جهت عمودی لبه ها smooth شده اند.

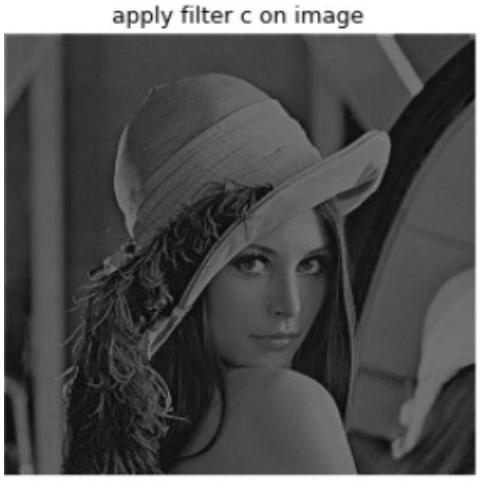
شکل۹)نتیجه اعمال فیلتر a_2

طبق شکل ۵ و ۹ میتوان نتیجه گرفت کـه فیلـتر a_2 در جهت افقی میانگین گیری کـرده اسـت چـون مقـداری در جهت افقی لبه ها smooth شده اند.



شکل۱۰)نتیجه اعمال فیلتر b

با توجه به شکل ۲ و ۱۰ این نکته تأکید می‌شود که فیلتر b یک فیلـتر highpass و edge detector لاپلاسـین است و به همین دلیل نتیجه اعمال بر تصویر که در شـکل ۱۰ نشان داده شده است لبه های افقی و عمودی و قطری تصویر را نشان میدهد.



شکل۱۱)نتیجه اعمال فیلتر c

مشاهده می‌شود که پس از اعمال فیلتر c در شکل ۱۱ لبه ها تقویت شده‌اند. به عنوان مثال با مقایسه کردن لبه های بالای کلاه در شکل ۱۱ و شکل ۶ میتوان این شارپ شدن و تقویت شدن لبه ها را مشاهده کرد و به این معنـا اسـت که فیلتر c یک فیلتر edge enhancement است.

۲-۲ تمرین ۲-۱-۴

در این تمرین خواسته شده است تا DFT را بر روی سه تصویر Lena, Barbara, F16, Baboon در حالت‌های با شیف و بدون شیفت و با لگاریتم و بدون لگاریتم اجرا کرده و نتایج تحلیل شود.
در بخش قبل بیان شد که علت شیف دادن تنها به علت مشاهده کردن بهتر طیف حوزه فرکانس به صورت بصری است, چون مرکز فرکانسی که دارای مقادیر بیشتری است را از بالا سمت چپ به مرکز مختصات شیفت میدهد وطیف بهتر مشاهده میشود.
وقتی طیف یا طیف توانی یک سیگنال یا تصویر در حوزه فرکانس رسم میشود, این طیف مقادیر بسیار متنوعی به عنوان مثال از ۰ تا ۱ میلیون دارد و مشاهده این مقادیر متفاوت به نحوی که از هم متمایز باشد برای چشم انسان ممکن نیست به همین علت از لگاریتم استفاده می‌شود تا این رنج وسیع را کاهش دهد و با مقدار دادن مقادیر تیره به رنج بیشتر کنتراست و کیفیت تصویر را افزایش دهد.
با استفاده از لگاریتم روی طیف, رنج زیاد مقادیر طیف کمتر می‌شود و میتوان طیف را بهتر دید.

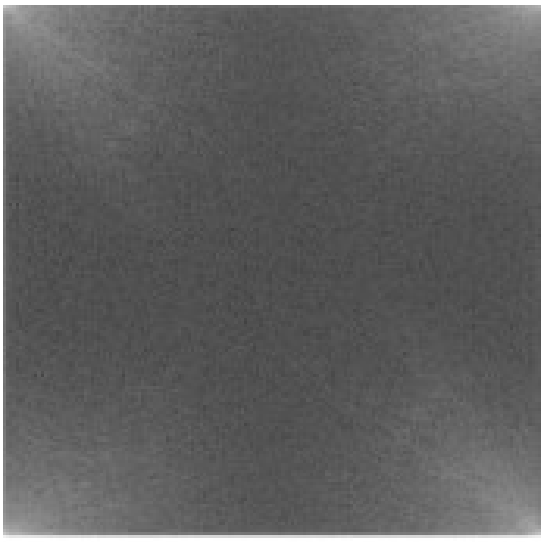
در تصاویر شیفت نخورده باید مرکز فرکانسی که مقادیر روشن(زیاد) دارد در چهار گوشه باشد و در تصویر شیفت خورده این مقادیر روشن در مرکز تصویر قرار میگرند طیف های بدون لگاریتم هم صفحات تیره هستند مه تنها در وسط آن‌ها یک نقطه روشن است به علت عدم امکان تمایز دادن بین بازه زیادی از شدت روشنایی اما در طیف با لگاریتم طیف به خوبی مشاهده میشود.

برای پیاده‌سازی لگاریتم از تابع log_transform که مربوط به تمارین سری ۲ است, استفاده شده است. (توضیحات این تابع به طور کامل در تمرین ۲ داده شده است). این تابه یک عکس را به عنوان ورودی دریافت کرده و لگاریتم آن را طبق فرمول تبدیل لگاریتم خروجی میدهد.

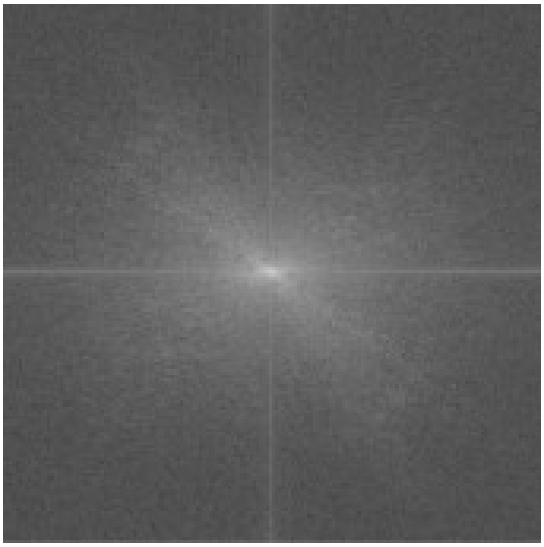
در تابع Q_4__1_2 کد مربوط به تمرین ۲-۱-۴ پیاده سازی شده است. این تابه به عنوان ورودی عکس را دریافت میکند و ۴ نسخه متفاوت را به عنوان خروجی میدهد.
ایتدا تنها از عکس ورودی تبدیل فوریه میگیرد و مقدار , magnitude آن را به عنوان که نه شیفت داده شده و نه لگاریتم از آن گرفته شده با عنوان mag خروجی میدهد.
خروجی دوم این تابع mag_shifted است که خروجی تبدیل را شیفت میدهد اما لگاریتم از آن گرفته نشده.
خروجی سوم تابع mag_log_است که پس از تبدیل فوریه گرفتن از خروجی لگاریتم میگیرد اما شیفت نخورده است.
خروجی چهارم تابع mag_log/_shifted است که بعد از تبدیل فوریه, شیفت مرکز فرکانسی صورت گرفته و بعد بر روی magnitude این خروجی تبدیل لگاریتم (لگاریتم) اعمال شده است.
نتایج این ۴ حالت بر روی ۴ عکس خواسته شده:

شکل۱۲)طیف lena بدون شیفت و بدون لگاریتم

شکل۱۳) طیف lena با اعمال شیفت و بدون لگاریتم.(به نقطه سفید کوچک وسط طیف دقت شود)



شکل۱۴)طیف lena بدون شیفت و با اعمال لگاریتم

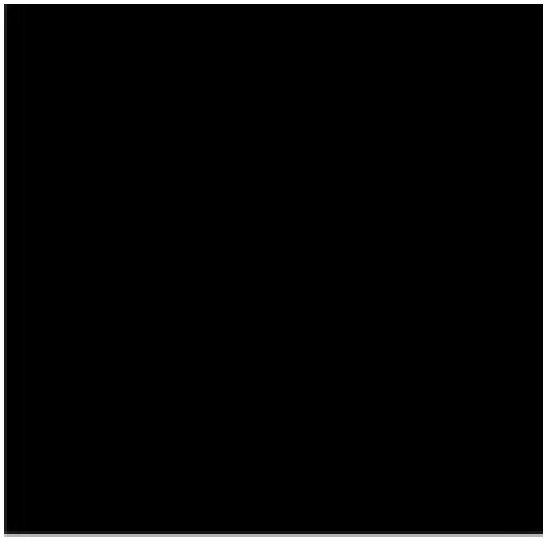


شکل۱۵)طیف lena با اعمال شیفت و اعمال لگاریتم

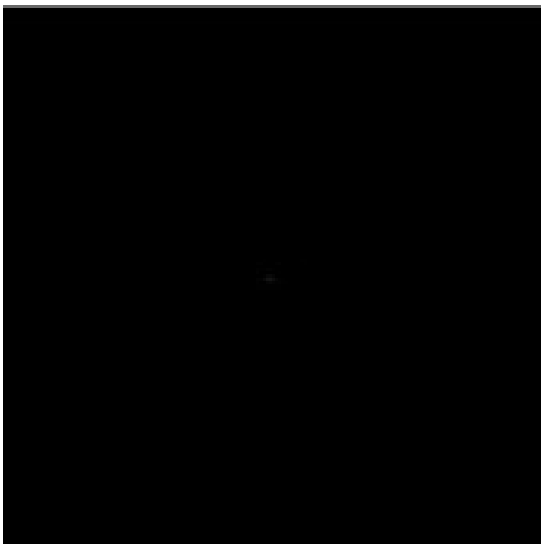
ابتدا نتایج طیف های lena را بررسی میکنیم. در شکل ۱۲ چون شیفت اعمال نشده مرکز فرکانسی در چهار گوشه قرار دارد اما چون لگاریتم اعمال نشده تا بتوان این بازه را بهتر مشاهده کرد, کل طیف به صورت یکپارچه سیاه مشاهده می‌شود .و نمیتوان از آن اطلاعاتی به دست آورد.
در شکل ۱۳ چون شیفت مرکز فرکانسی به مبدأ صورت گرفته در مرکز شکل ۱۳ یه دایره کوچک سفید رنگ(کمی باید دقت شود) دیده میشود. این نقطه حاصل جمع مقادیر تمامی پیکسل ها میباشد.
در شکل ۱۴ چون لگاریتم گرقته شده است و رنج بسایر زیاد طیف توسط تبدیل لگاریتمی به رنج محدود تری تبدیل شده است و نقاط تیره کمتراست بیشتری پیدا کرده‌اند طیف را بهتر میتوان دید. چون شیف مبدأ فرکانسی داده نشده مشاهده می‌شود که مبدأ فرکانسی در چهار گوشه عکس روشن است.
در شکل ۱۵ هم لگاریتم اعمال شده و هم شیفت بمدا فرکانسی و مشاهده می‌شود که مبدأ فرکانسی در وسط عکس سفید است. این نقطه حاصل جمع مقادیر تمامی پیکسل ها میباشد.
این نتایج کلی از ۴ طیف lena برای طیف های عکس های Barbara, F16, Baboon که در ادامه آورده می‌شود نیز به همین صورت میباشد وبرای جلوگیری از تکرار دیگر تکرار نمیشود.

نکته دیگری که در شکل ۱۵ میتوان مشاهده کر این است که در این طیف, خط ها روشن عمودی افقی وجود دارند که به ترتیب نشان دهنده لبه افقی و عمودی در تصویر lena هستند همچنین خط روشن دیگریدر طیف مه در زاویه ۱۳۵ درجه از محور مختصات قرار دارد نشان دهنده لبه ۴۵ درجه در عکس lena میباشد.(لبه های دور کلاه در عکس ۴۵ درجه هستند) . همچین ضرایب فوریه برای لبه های افقی و عمودی تقریباً مشابه هستند در حالی که برای ۱۳۵ درجه حوزه فرکانس(۴۵ درجه حوزه مکان) مقادیر این ضرایب متفادت است.

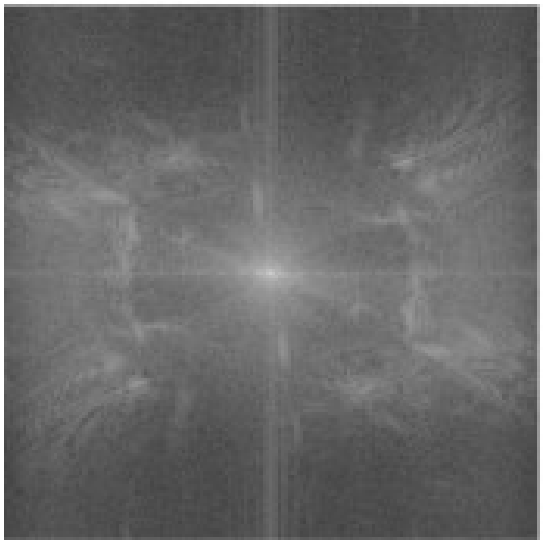
نتایج ۴ حالت طیف Barbara:



شکل۱۶)طیف Barbara بدون شیفت و بدون لگاریتم



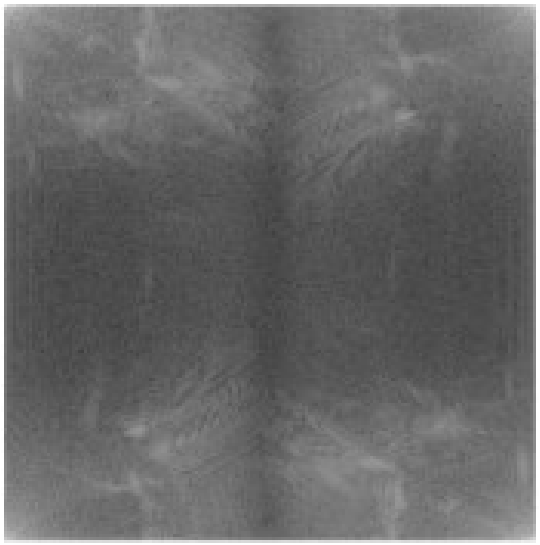
شکل۱۷)طیف Barbara با اعمال شیفت و بدون لگاریتم.(به نقطه سفید کوچک وسط طیف دقت شود)



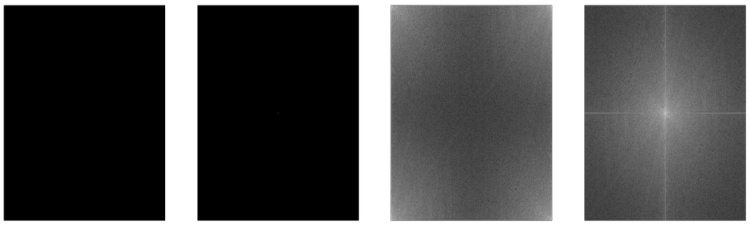
شکل۱۹)طیف Barbara با اعمال شیفت و با اعمال لگاریتم

در شکل ۱۶ و ۱۷ که ورژن های بدون لگاریتم هستند, طیف های حاصل همانند شکل ۱۲ و ۱۳ برای lena هستند. در شکل ۱۹ مقادیر زیادی در خط عمودی ویط طیف مشاهده می‌شود که نشان دهنده لبه های افقی کوچک در تصور حوزه مکان(برعکس بودن حوزه فرکانس و زمان) میباشد. این لبه های افقی کوچک در شلوار موجود در عکس مشاهده میشوند.

نتایج ۴ حالت طیف F۱۶:



شکل۱۸)طیف Barbara بدون شیفت و با اعمال لگاریتم



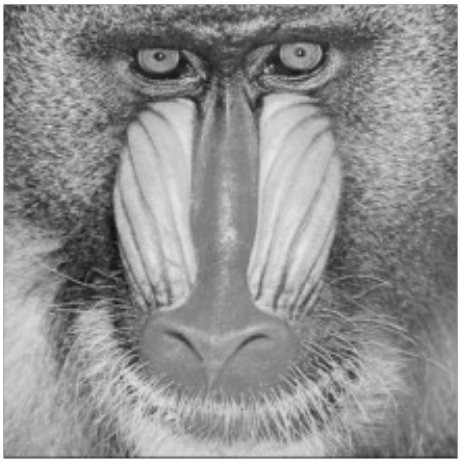
شکل از سمت چپ ۲۰و۲۱و۲۲و۲۳) طیف های مربوط به F16

درورژن های بدون لگاریتم مانند شکل ۲۰ و ۲۱ مشاهده می‌شود که نتایج برایF16 مانند تصاویر لنا و باربارا است. در شکل ۲۳ مشاهده می‌شود که مقادیر محور عمودی بزرگ‌تراز محور افقی هستند. این به این معناست کعه تصویر F16 در حوزه مکان لبه های افقی مشخص تری نسبت به لبه های عمودی است. که کاملاً نیز لبه افقی خود بدنه هواپیما این را تأیید میکند.

نتایج ۴ حالت طیف Baboon:

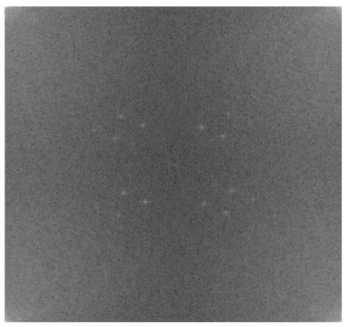
شکل۲۴)تصویر *Barbara*

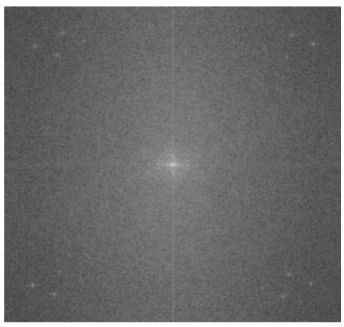
شکل۲۵)تصویر *F16*



شکل۲۶)تصویر *Baboon*

شکل۲۷و۲۸ ازچپ)طیف Baboon بدون لگاریتم و بدون شیفت - طیف Baboon بدون لگاریتم و با اعمال شیفت





شکل۲۹و۳۰ازچپ)طیف Baboon با لگاریتم و بدون شیفت - طیف Baboon با لگاریتم و با اعمال شیفت

در شکل ۳۰ در طیف Baboon مشاهده می‌شود که تجمع مقادیر بالا فرکانس در مرکز میباشد و همچنین در هر چهار گوشه طیف نیز سه نقطه با مقدار فرکانسی بالا وجود دارد. این به این دلیل است که در تصویر Baboon و به‌خصوص در قسمت صورت, طرح هایی که در صورت دیده می‌شود مانند نویز است و همچنین به طور کلی است تصویر تغییراتی با فرکانس بالا دارد که این تغییرات غیر پریودیک هستند و این سه نقطه مقدار بالا در هر چهار جهت طیف شکل ۳۰ دلالت بر این موضوع دارد.

۲-۳ تمرین ۱-۲-۴

در قسمت اول گفته شده که عکسی با سایز ۲۵۶ در ۲۵۶ داده شده است و هدف آن است که خروجی فیلتر در حوزه زمان و هم فرکانس هم در سایز ۲۵۶ در ۲۵۶ باشد. میدانیم در حوز همکان برای اینکه سایز تصویر ورودی با خروجی حاصل از فیلتر باربر باشد باید از same padding استفاده کرد.same padding به این‌گونه است

که اگر سایز عکس n* n و سایز فیلتر f*f باشد, سایز پدینگ در هر لبه عکس باید ۲/(۱-f) باشد.
در اینجا گفته شده که سایز فیلتر ۱۱ است. طبق فرمول بالا پدینگ باید ۱۰ باشد.این پدینگ ۱۰ از هر سمت به تصویر اضافه میشود, پس سایز تصویر به ۲۵۶ +ـ ۲(۱۰) که برابر با ۲۷۶ میرسد. در نتیجه در بخش اول این سؤال باید فیلتر پدینگ شوند تا به سایز ۲۷۶ برسند و بعد در حوزه فوریه ضرب ارایه ای شوند. در این صورت سایز خروجی فیلتر در فرکانس با سایز خروجی فیلتر در مکان با سایز تصویر اصلی برابر خواهد بود.

در قسمت بعدی مراحل convolution خواسته شده است که به صورت زیر است:
۱- سایز تصویر اگر m درn است باید به نحوی pad شود که سایز آن ۲m در ۲n شود
۲- شیفت صورت بگیرد که مرکز فرکانسی روی مرکز مختصات بیفتد برای نمایش بهتر طیف. این کار به دور ش ممکن است استفاده از fftshift بعد از تبدیل فوریه یا ۱- به توان x+y در حوزه مکان.
۳- تبدیل فوریه در مسیر رفت
۴- ضرب آرایه ای فیلتر در حوزه فرکانس در عکی در حوزه فرکانس
۵- معکوس تبدیل ۲ ifft
۶- شیفت دادن مرکز فرکانسی از مرکز مختصات به ۰و ۰

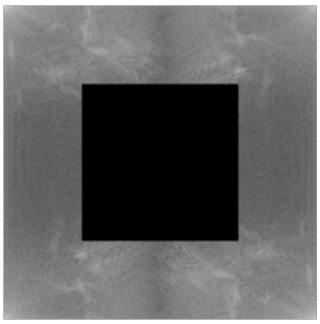
برای قسمت بعدی سؤال, گفته شده که ۲۵۶ در ۲۵۶ نقطه در DFT استفاده شده است و این به این معنی است که خروجی IDFT در برگشت به حوزه مکان از فرمانس پس ار اعمال فیلترینگ ۲۵۶ در ۲۵۶ است و از آنجایی که میدانیم در مراحل فلتریگ حوزه فرکانس طول و عرض عکس ورودی دو برابر می‌شود (طبق مراحل بالا) پس اندازه عکس ورودی ۱۲۸ در ۱۲۸ است. M = n = ۱۲۸

۴-۲-۴ تمرین ۲-۲-۴

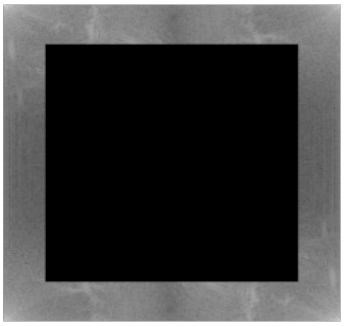
در این تمرین خواسته شده است تا ابتدا تبدیل فوریه گسسته برای عکس بارابارا محاسبه شود و بعد ضرایب فوریه با شرایط قسمت‌های a و b صفر شوند و بعد با معکوس تبدیل عکس به حوزه مکان بازگردد و normalize صورت بگیرد و نتایج بررسی شوند.

تابع normalize عملیات نرمالایز کردن را انجام میدهد به این صورت که ورودی یک عکس را دریاف میکند و با به دست آوردن مقدار ماکزیمم و مینمم در این عکس با استفاده از np.mean و np.max و استفاده از فرمول نرمالایز کردن خطی مقادیر پیکسل هارا نرمال میکند بین ۰ و ۲۵۵.

بررسی قسمت a :
برای این قسمت تابع Q_۴_۲_۲_a پیاده‌سازی شده است.
این تابع برای ورودی عکس و مقدار Tرا دریافت میکند. ابتدا عکس را به حوزه فوریه میبرد و پس از به دست آوردن ضرایب فوریه نواحی که در قسمت a گفته شده است را در ناترسی ضرایب فوریه برابر ۰ قرار میدهد و با معکوس تبدیل فوریه به مکان برمیگردد و خروجی را نرمال میکند با تابع normalize و خروجی نرمالایز شده و خروجی را در حوزه فرکانس را خروجی میدهد.

شکل۳۱)اعمال شرایط a با T=1/4

شکل۳۲)اعمال شرایط a با T=1/8

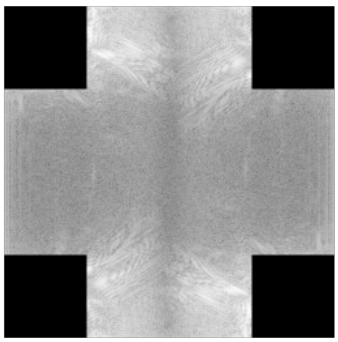
در پیاده‌سازی تایع قسمت a از fftshift استفاده نشده استدر نتیجه در وسط طیف فرکانس های بالا وجود دارند. در شکل ۳۱ و ۳۲ باتوجه به طیف دیده می‌شود که قسمت‌های مرکز با توجه به مقدار T مقدار ۰ دارند. در T برابر با ۱/۸در شکل ۳۲ تاحیه مشکی بزرگ‌تر است.
بر اساس شکل ۳۲ و ۳۲ بر اساس حذف کردن مقادیر در قسمت a فرکانس های بالا که در مرکز قرار دارند و شامل لبه ها و نویز های تصویر هستند ۰ می‌شوند. با ۰ کردن این مقادیر درواقع همانند فیلتر lowpass رفتار می‌شود و تصویر smooth میشود. در شکل ۳۲ چون ناحیه ۰ شده بزرگ‌تر است و فرکانس های بالای بیشتری را عبور میدهد در نتیجه تصویر نیز بیشتر از شکل ۳۱ smooth

شده است. این مضوع را میتوان با مقایسه لبه های آرنج و دست‌ها در دو تصویر ۳۱ و ۳۲ بیشتر تشخیص داد.

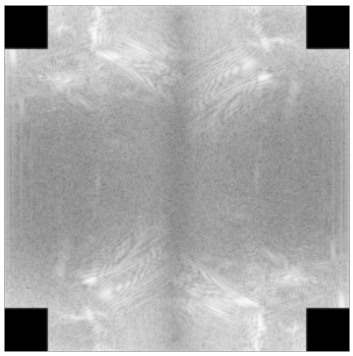
برای شرط b نیز تابع Q_4_2_2_b پیاده‌سازی شده اینکه همانند تابع قسمت a است با این تفاوت که شرط هایی که در قسمت b داده شده‌اند درواقع ۴ قسمت گوشه طیف را ۰ میکنند و این تابع پس از تبدیل کردن تصویر ورودی به حوزه فرکانس در ماتریس ضرایب فوریه با توجه به مقدار T که در ورودی دریافت کرده است, ضرایب که در این ۴ گوشه قرار دارند را ۰ میکند و با معکوس تبدیل به حوزه مکان باز میگردد و خروجی را نرمالایز میکند و این خروجی نرمالایز شده و مقدار خروجی در حوزه فرکانس را خروجی میدهد.

فرکانس های پایین را عبور نمیدهد و فرکانس های بالارا عبور میدهد.

بر اساس شروط b شکل ۳۳ که مقدار T در آن برابر با ۱/۴ است, سطح گوشه‌هایی که مقدار ۰ میگیرند(فرکانس ها پایین) در طیف بیشتر است از شکل ۳۴. به این معنا که در شکل ۳۳ ضرایب بیشتری از شکل ۳۴ حذف می‌شود و یعنی در شکل ۳۳ چون محدوده بیشتری حذف می‌شود عملاً مقداری فرکانس های پایین‌تری را هم ۰ میکند. نتیجه این استکه شکل ۳۳ از شکل ۳۴ بیشتر کلیات را حذف میکند. این نکته را میتوان با توجه به سطح پشت فرد در عکس که در ۳۴ هنوز اثراتی از آن هست اما در ۳۳ کاملاً از بین رفته است بیشتر متوجه شد.

شکل۳۳)اعمال شرایط b با T=1/4

شکل۳۴)اعمال شرایط b با T=1/8

در پیاده‌سازی تایع قسمت b از fftshift استفاده نشده استدر نتیجه در ۴ گوشه طیف فرکانس های پایین (کلیات تصویر) وجود دارند.
طبق شروط bضرایب فوریه نواحی در ۴ گوشه طیف ۰ می‌شوند و چون در این نواحی فرکانس های پایین و کلیات تصویر وجود دارند درواقع شروطb برخلاف a با ۰ کردن این نواحی مانند فیلتر highpass عمل میمند چون

۳- کد تمرینات

```python
# -*- coding: utf-8 -*-
"""Frequency_Domain_Sara_Ghavampour_9812762781.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/11zSfG3WFScbFX2w5zWnQRZ2H7qQ7yGYg
"""

!wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --no-check-certificate

!wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --no-check-certificate

!wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --no-check-certificate

!wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --no-check-certificate

# Commented out IPython magic to ensure Python compatibility.
import numpy as np
import matplotlib.pyplot as plt
import sklearn
import cv2
from math import *
from sklearn.metrics import mean_squared_error
# %matplotlib inline

def show_img(*args, figsize=10, is_gray=True, title=None, fontsize=12):
    if isinstance(figsize, int):
        figsize = (figsize, figsize)
    images = args[0] if type(args[0]) is list else list(args)
    cmap=None
    if not is_gray:
        images = list(map(lambda x: cv2.cvtColor(x, cv2.COLOR_BGR2RGB), images))
    else:
        cmap = 'gray'
    plt.figure(figsize=figsize)
    for i in range(1, len(images)+1):
        plt.subplot(1, len(images), i)
        if title is not None:
            plt.title(title[i-1], fontsize=fontsize)

        plt.imshow(images[i-1], cmap=cmap)
        plt.axis('off')

lena_img=cv2.imread('Lena.bmp')
lena_img = cv2.cvtColor(lena_img,cv2.COLOR_BGR2GRAY)
plt.imshow(lena_img,cmap='gray')
plt.axis('off')
plt.show()

"""4.1. fourier transform

4.1.1
"""

###------------------------- 4.1.1

def pad_before_fft(img,pad_size):
  padded_img = np.zeros((2*pad_size,2*pad_size))
  padded_img[0:img.shape[0],0:img.shape[1]] = img
  return padded_img

show_img(pad_before_fft(lena_img,512))

"""plotting magnitude"""

def fourier_magnitude(x,pad_size):
  x_pad=pad_before_fft(x,pad_size)
  fft=np.fft.fftshift(np.fft.fft2(x_pad))
  return np.abs(fft)

filter_a=np.array([[1,2,1],[2,4,2],[1,2,1]])*(1/16)

show_img(fourier_magnitude(filter_a,512))

filter_a_1 = (1/4)*np.array([[1],[2],[1]])
show_img(fourier_magnitude(filter_a_1,512))

filter_a_1

filter_a_2 = (1/4)*np.array([[1,2,1]])
show_img(fourier_magnitude(filter_a_2,512))

filter_b=np.array([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])
show_img(fourier_magnitude(filter_b,512))

filter_c = np.array([[0,-1,0],[-1,5,-1],[0,-1,0]])
show_img(fourier_magnitude(filter_c,512))

"""apply filters in frequency domain"""

def filter_frequency_domain(img,filter):
  img_pad = pad_before_fft(img,img.shape[0])
  x_fft=np.fft.fftshift(np.fft.fft2(img_pad))

  filter_pad = pad_before_fft(filter,img.shape[0])
  filter_fft=np.fft.fftshift(np.fft.fft2(filter_pad))

  out = x_fft * filter_fft
  # return np.fft.ifft2(np.fft.ifftshift(out))
  return abs(np.fft.ifft2(np.fft.ifftshift(out)))[0:img.shape[0],0:img.shape[1]]

freq_filtered_a_1 = filter_frequency_domain(lena_img,filter_a_1)
show_img(lena_img,freq_filtered_a_1,title=['original image','apply filter a_1 on image'])

freq_filtered_a_2 = filter_frequency_domain(lena_img,filter_a_2)
show_img(lena_img,freq_filtered_a_2,title=['original image','apply filter a_2 on image'])

freq_filtered_a = filter_frequency_domain(lena_img,filter_a)
show_img(lena_img,freq_filtered_a,title=['original image','apply filter a on image'])

freq_filtered_b = filter_frequency_domain(lena_img,filter_b)
show_img(lena_img,freq_filtered_b,title=['original image','apply filter b on image'])

freq_filtered_c = filter_frequency_domain(lena_img,filter_c)
show_img(lena_img,freq_filtered_c,title=['original image','apply filter c on image'])

"""4.1.2"""

###----------------- 4.1.2
```

```
def log_transform(img):  # s = c log(r+1)   # 255/log(r_max +1)
  r_max = np.max(img)
  c = 255/log(r_max +1)
  out = img.copy()
  out = c * np.log(img+1)
  return out.astype('int32')

def Q_4_1_2(x):
  mag=abs(np.fft.fft2(x)) # without shift- without log
  mag_shifted=abs(np.fft.fftshift(np.fft.fft2(x))) # with shift- without log
  mag_log=log_transform(abs(np.fft.fft2(x))) # without shift- with log
  mag_log_shifted=log_transform(abs(np.fft.fftshift(np.fft.fft2(x)))) # with shift- with log
  return mag,mag_shifted,mag_log,mag_log_shifted

plt.imshow(lena_img,cmap='gray')
plt.axis('off')
plt.show()

# 4.1..2 on lena
lena_mag,lena_mag_shifted,lena_mag_log,lena_mag_log_shifted=Q_4_1_2(lena_img)#
show_img(lena_mag,lena_mag_shifted,lena_mag_log,lena_mag_log_shifted,figsize=35)

barbara_img=cv2.imread('Barbara.bmp')
barbara_img = cv2.cvtColor(barbara_img,cv2.COLOR_BGR2GRAY)
plt.imshow(barbara_img,cmap='gray')
plt.axis('off')
plt.show()

# 4.1..2 on barbara
barbara_mag,barbara_mag_shifted,barbara_mag_log,barbara_mag_log_shifted=Q_4_1_2(barbara_img)
show_img(barbara_mag,barbara_mag_shifted,barbara_mag_log,barbara_mag_log_shifted,figsize=35)

F16_img=cv2.imread('F16.bmp')
F16_img = cv2.cvtColor(F16_img,cv2.COLOR_BGR2GRAY)
plt.imshow(F16_img,cmap='gray')
plt.axis('off')
plt.show()

# 4.1..2 on f16
F16_mag,F16_mag_shifted,F16_mag_log,F16_mag_log_shifted=Q_4_1_2(F16_img)
show_img(F16_mag,F16_mag_shifted,F16_mag_log,F16_mag_log_shifted,figsize=35)

Baboon_img=cv2.imread('Baboon.bmp')
Baboon_img = cv2.cvtColor(Baboon_img,cv2.COLOR_BGR2GRAY)
plt.imshow(Baboon_img,cmap='gray')
plt.axis('off')
plt.show()

# 4.1..2 on Baboon
Baboon_mag,Baboon_mag_shifted,Baboon_mag_log,Baboon_mag_log_shifted=Q_4_1_2(Baboon_img)
show_img(Baboon_mag,Baboon_mag_shifted,Baboon_mag_log,Baboon_mag_log_shifted,figsize=35)

"""4.2

4.2.2
"""

#---------------------- 4.2.2

def normalize(img):
  min = np.min(img)
  max = np.max(img)
  return ((img-min)/(max-min) * 255).astype('uint8')

#----------------- part a

def Q_4_2_2_a(img,T):
  n = img.shape[0]
  dft = np.fft.fft2(img)
  filter_dim = int((1-T)*n) - int(T*n)
  dft[int(T*n):int((1-T)*n) ,int(T*n):int((1-T)*n) ] = np.zeros((filter_dim,filter_dim))
  out = np.abs(np.fft.ifft2(dft))
  out_freq= np.log(np.abs(dft))
  out_freq[out_freq == -inf] = 0
  return normalize(out),out_freq

# filter a , t = 1/4
temp,dft = Q_4_2_2_a(barbara_img,1/4)
show_img(dft,temp)

# filter a , t = 1/8
temp,dft = Q_4_2_2_a(barbara_img,1/8)
show_img(dft,temp)

#----------------- part b

def Q_4_2_2_b(img,T):
  n = img.shape[0]
  dft = np.fft.fft2(img)
  tn = int(T*n)
  t_1_n = int((1-T)*n)
  dft[0:tn ,0:tn] = np.zeros((tn,tn))
  dft[0:tn ,t_1_n:n] = np.zeros((tn,tn))
  dft[t_1_n:n ,0:tn] = np.zeros((tn,tn))
  dft[t_1_n:n ,t_1_n:n] = np.zeros((tn,tn))
  out = np.abs(np.fft.ifft2(dft))
  out_freq= np.log(np.abs(dft))
  out_freq[out_freq == -inf] = 0
  return normalize(out),out_freq

# filter b , t = 1/4
temp,dft = Q_4_2_2_b(barbara_img,1/4)
show_img(dft,temp)

# filter b , t = 1/8
temp,dft = Q_4_2_2_b(barbara_img,1/8)
show_img(dft,temp)
```