

HW3

حامد محمدزاده ۹۸۴۷۶۲۴۱۸

توضیح تبدیل به حوزه فرکانسی:

از numpy و ماژول fft استفاده میکنیم.
توضیح تکایع:

```
def get_mag(img):  
    f = np.fft.fft2(img)  
    f = np.fft.fftshift(f)  
    mag = np.abs(f)**2  
  
    return mag
```

با استفاده از این تابع
magnitude spectrum

تصویر را نمایش میدهیم.

موضوع این تابع فقط برای انجام تغییرات در

طیف استفاده می شود. در این منحنی و نه image استفاده نمی شود.

اندازه کار:

ابتدا تبدیل عکس به فرکانس را انجام دادیم، پس فرکانسها
را به عدد برگردانیم. در نهایت مقدار real را برمی داریم و در آن طیف
تفسیر/استخراج می شود. (با تابع abs)

```
def display_mag(img):  
    f = np.fft.fft2(img)  
    fshift = np.fft.fftshift(f)  
    #return np.abs(fshift)  
    magnitude_spectrum_display = 20*np.log(np.abs(fshift))  
    return magnitude_spectrum_display
```

این تابع برای visualization گرفتن طیف تغییر استفاده می شود،
تنها مقادیر که با `get_mag()` وارد می شود که در مقایسه طیف
نگاریم. مگر در آن توضیح اکنون بیشتر شود.

```
def get_phase(img):  
    dft = np.fft.fft2(img)  
    dft_shift = np.fft.fftshift(dft)  
    phase_spectrum = np.angle(dft_shift)  
    return phase_spectrum
```

این تابع برای گرفتن phase
تفسیر استفاده می شود.

```
def mask(img, filter_f):  
    f = np.fft.fft2(img)  
    fshift = np.fft.fftshift(f)  
  
    fshift_res = fshift * filter_f  
  
    f_ishift = np.fft.ifftshift(fshift_res)  
    inv_img = np.fft.ifft2(f_ishift)  
    inv_img = np.abs(inv_img)  
  
    return inv_img
```

این تابع به این صورت عمل می کند که تصویر را به همراه فیلتر مورد نظر (فیلتر در فضای فرکانسی است)
می گیرد، تصویر را به حوزه فرکانسی می برد و فیلتر را اعمال می کند، پس تصویر اصلاح شده را مجدداً به حوزه مکان
برگرداند.

منوریم به این معنی که در تصویر اعمال می کنیم، فیلترها می تواند فیلتر را اعمال می کند،

در یک عکس فیلتر را انجام داده و inverse fourier transform می کنیم. در نهایت به آنهایی که می گذارند

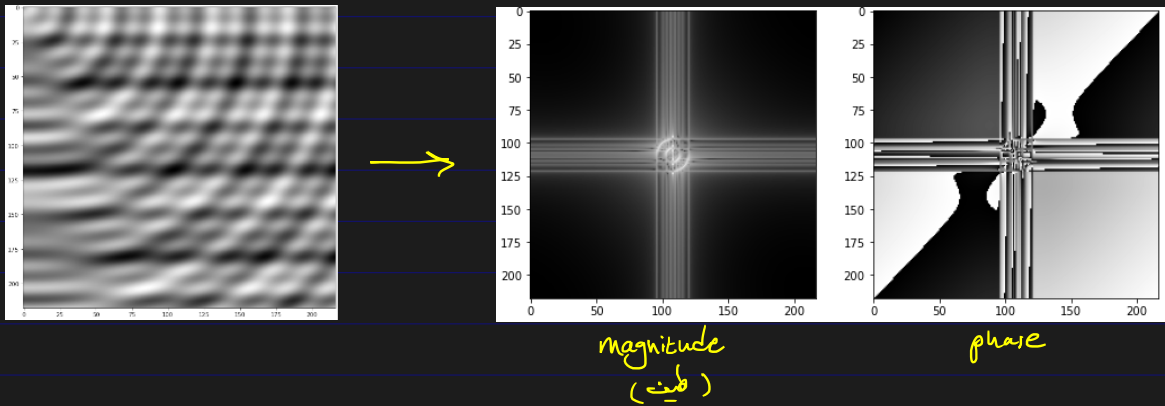
در اگر کم طویلی مقدار صفر می شود و بعداً طیف $abs()$ را می گیرد تا مقایسه با $abs()$ داشته باشد.

$$\underbrace{\sin(0.1n) + \sin(0.2n) + \cos(0.4n)}_{\text{I}} + \underbrace{\sin(\sqrt{2} \times 0.1n) + \sin(\sqrt{2} \times 0.2n)}_{\text{II}}$$

سوال ۱:

- تفسیر هر دو چگ که از رابطه گفته شده از سوال به دست می آید در زیر آمده است:

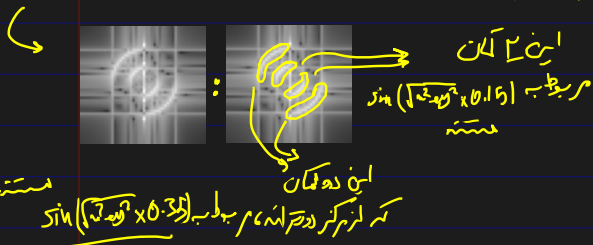
هر زمانه افقی که در شکل به مدار periodic شکل گرفته، ناشی از روابط مثلثاتی (I) است.
هر زمانه که به طور دایره وار کند گرفته شده، ناشی از روابط مثلثاتی (II) هستند.



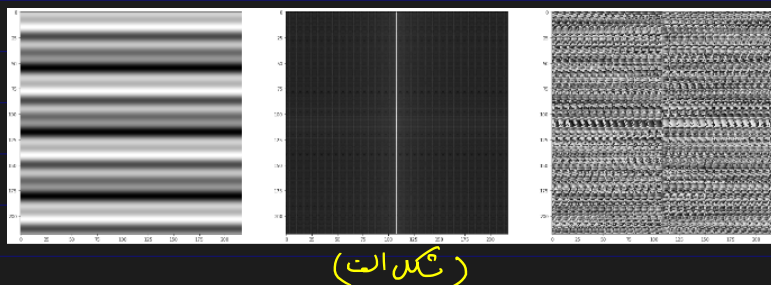
برای طیف تفسیر:

(برای تفسیر این حوضه ها، مکان در حوضه ها را می بینیم، محور است)
هر زمانه افقی که periodic هستند، در طیف به صورت یک خط عمودی که از وسط طیف می گذرد نمایان می شود. (شکل الف)
هر زمانه دایره وار از π شروع به طیف می کنند: ① این هر زمانه یک مولفه $\sqrt{2}$ دارند که در طیف به صورت خطی عمودی مشخص می شود. (شکل ب)
② این هر زمانه یک مولفه $\sqrt{2}$ دارند که به صورت خطی افقی در طیف مشخص می شود.

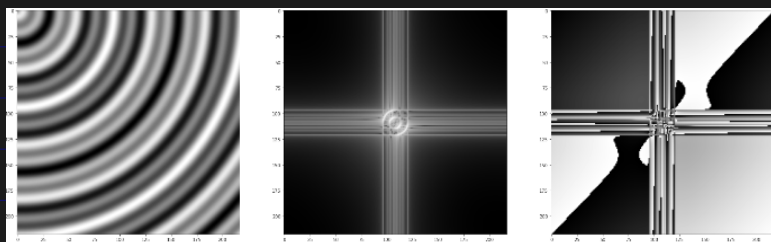
③ روابط (II) به صورت یک مربع $\sqrt{2}$ دارند و در طیف به صورت $\frac{1}{2}$ تا ربع دایره مشخص می شود.



اگر روابط مثلثاتی (I) و (II) را از هم جدا کرده، طیف آن را رسم کنیم، این موضوع به وضوح دیده می شود:

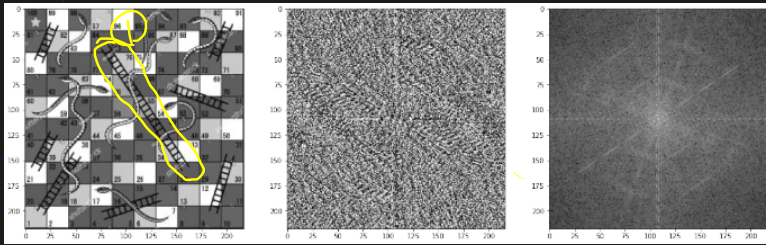


(شکل الف)



(شکل ب)

سوال ۲)



phase

magnitude

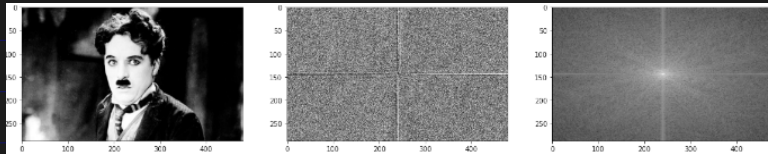
بررسی طیف تصویر سارایله :

① این لاینر نشان می‌دهد که (۱) است.

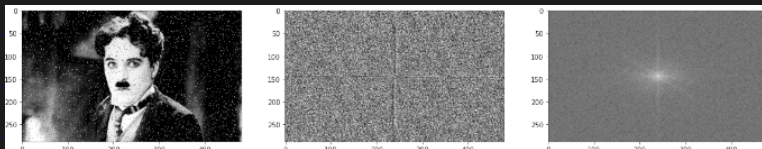
② (۴) این خطوط طیف، اثر مرزهای افقی و عمودی است که تایم در آن منعقد می‌شود. پس باید دور آن‌ها.

④ این خطوط طیف که تقریباً دایره‌ای هستند، مربوط به لاینر ماژیک (که فرد دیده‌اند) در طیف است.

سوال ۳)

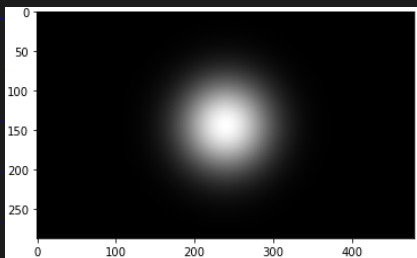


قبل از اعمال نویز salt and pepper



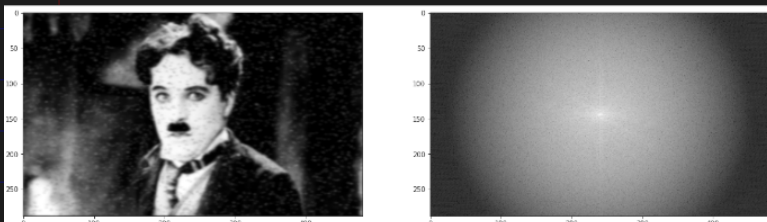
بعد از اعمال نویز:

بعنوان آنکه دامنه است، طیف زگانی در مرکز بهر دست در نویز pepper است. پس این دستگانی‌ها با لاینر تصویر شده.



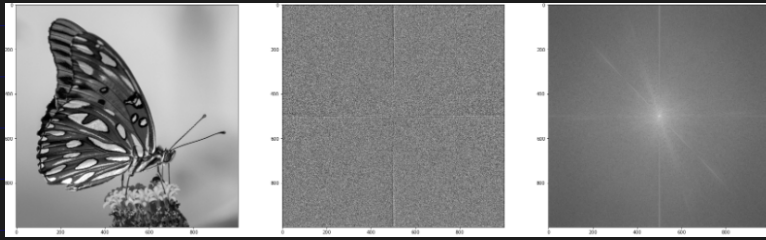
کارنامه می‌توان انجام داد، smooth کردن تصویر با low pass gaussian است:

از آنجمله نویز periodic است، پس باید آن را band reject (بند دور) کرد.

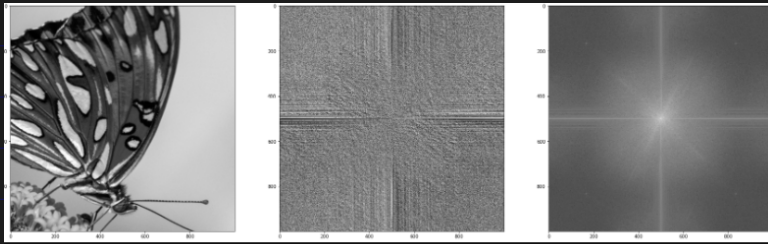


تصویری smooth شد با low pass gaussian:

سوال 4

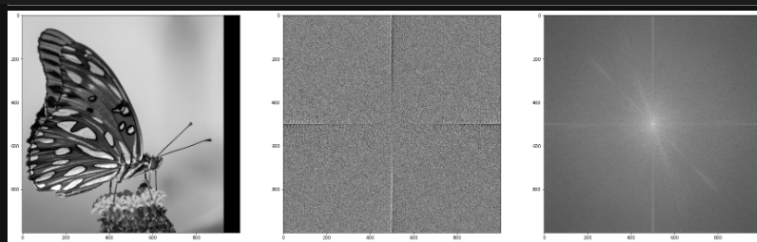


a) طیف تصویر در حوزه فرکانس:



b) طیف تصویر بعد از حذف:

واضح است که طیف تصویر با حذف برخی تغییر مکان (چون زایه لبه ها تغییر مکانه) (تغییر مکان)



c) طیف بعد از حذف:

واضح است که طیف تصویر با حذف تغییر مکان

سوال 5

```
def ideal_br(img, d0, w):
    bf = np.ones((img.shape))
    cx = img.shape[0]/2
    cy = img.shape[1]/2

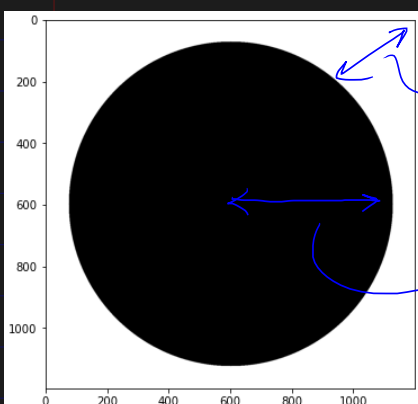
    for x in range(bf.shape[0]):
        for y in range(bf.shape[1]):
            d = np.sqrt((x-cx)**2 + (y-cy)**2)
            if (d >= d0 and d <= d0 + w):
                bf[x][y] = 0

    return bf
```

این تابع با فرض داشتن d_0 به عنوان cutoff frequency و w به عنوان width
 یک فیلتر ایده‌آل $band\ reject$ را می‌سازد

- برای اینکه فیلتر $band\ reject$ بتواند فقط لبه‌ها را نگه دارد، باید مرکز آن به مرکز تصویر باشد.

در d_0 را کم و w را افزایش دهیم، فیلتر حامل تقریباً $pass\ band$ است:

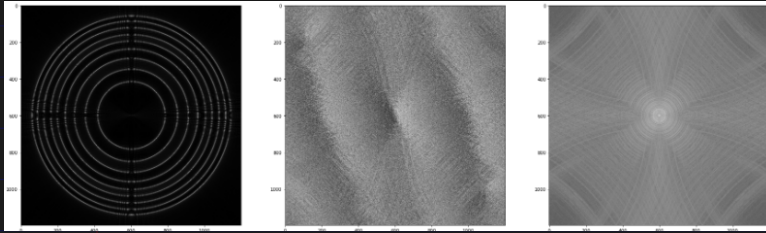


لبه (لبه) مرکز آن به مرکز تصویر باشد

مرکز آن به مرکز تصویر باشد
 لبه فیلتر را عبور
 یعنی دقت

اداره سوال (5)

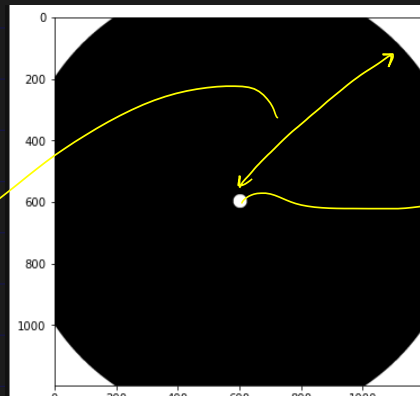
نتیجہ اعمال فیلتر بالابہ بقصر نہ حرزہ فرما کسی:



ہاں لور کہ واضح است
مقلد لبہ ما حقہ عمر لہ

سوال (6)

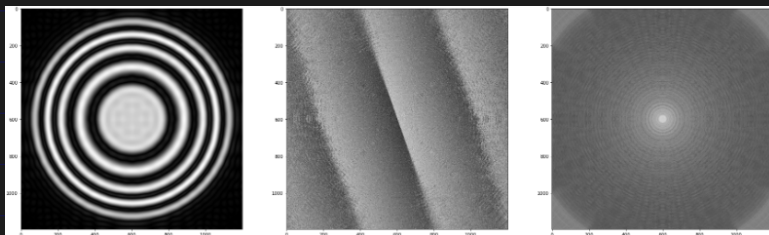
برہن لکہ مقابلہ ہمارا حذف کتہ، بلکہ معوضہ کارن ہاں یاسین را میسر دہیم۔
ہلکہ یہ را فقہ لیسہ ہاں۔
کافر کاسی ہاں بالارا عبور نہدہ۔
(لبہ نا)



خاکہ ہاں
بابا (لبہ نا) را
forget ماکہ

فیلتر:
فرکان یاسین
(آلیات تکرار) را عبور دہیم

نتیجہ:



بہا نظر کہ واضح است، لبہ نا smooth سہلہ.