

## فاطمه باغخانی

اطلاعات گزارش	Fourier Transform چکیده
<p>تاریخ: 99/2/12</p> <p><b>واژگان کلیدی:</b> Transform Domain Fourier Transform Inverse Fourier transform Frequency Spectra Uncertainty Relation فیلترینگ در حوزه فرکانس فیلتر بالاگذر فیلتر پایین گذر فیلتر میان گذر</p>	<p>ما در حوزه پردازش تصویر دو حوزه داریم حوزه مکان و حوزه تبدیل ما در حوزه مکان مستقیماً با پیکسل ها کار داریم اما در حوزه تبدیل پیکسل ها را مستقیماً دستکاری نکرده . برای پردازش تصویر در این حوزه ابتدا تصویر را به حوزه تبدیل برده و سپس فیلتر مورد نظرمان را اعمال میکنیم و سپس آن را بحوزه مکان برمیگردانیم برای تبدیل و بردن تصویر به حوزه فرکانس از تبدیل فوریه استفاده میکنیم .</p>

و نحوه پردازش در حوزه فرکانس  
میپردازیم

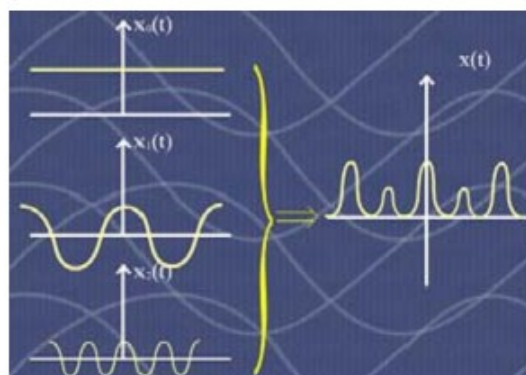
### 1-مقدمه

در این گزارش به پردازش در حوزه تبدیل میپردازیم تبدیل یک عمل ریاضی است که با دریافت یک تابع و یا رشته آن را به تابع و رشته دیگر تبدیل میکند علت استفاده در حوزه تبدیل این است که یک عمل مانند کانولوشن در این حوزه راحت تر اعمال می شود امکان فشرده سازی دیتا وجود دارد امکان اینکه اطلاعات مخفی در تابع را در این حوزه میتواند آشکار کند. ایده اولیه Fourier Transform این بود که هر تابع پریودیک را میتوان به صورت حاصل جمع از توابع سینوسی نوشت مانند شکل زیر که در ادامه به بررسی این تبدی

### 2-شرح تکنیکال

#### • Fourier Transform

در تبدیل فوریه سیگنال سینوسی و کسینوسی به عنوان توابع پایه ای استفاده می شوند. در واقع توابع نمایی هم شکل دیگر توابع سینوسی و کسینوسی است. بنابراین تبدیل فوریه یک سیگنال در بعد زمان یا مکان را به مجموعه ای از توابع سینوسی با فرکانس های مختلف تجزیه می کند. برای بازسازی سیگنال اصلی از روی تبدیل فوریه، چون توابع پایه ای ثابت می باشد (سینوسی)، تنها نیاز به دانستن مجموعه فرکانس هایی است که سیگنال اصلی شامل بوده است. بنابراین در اصل تبدیل فوریه نشان می



دهد که سیگنال شامل چه فرکانس هایی می باشد

رابطه تبدیل فوریه گسسته در دو بعد به صورت زیر است:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

و رابطه عکس تبدیل فوریه دو بعدی به صورت زیر است:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

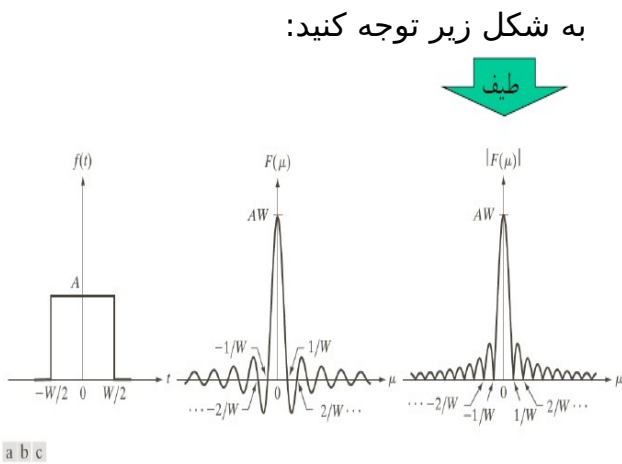


FIGURE 4.4 (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.

و تغییراتی که در طیف سیگنال انجام میگیرد را ملاحظه میکنید.

مشاهده می شود که تبدیل فوریه تصویری با ابعاد تصویر اصلی است.

تبدیل فوریه دارای مقدار حقیقی R و موهومی I می باشد، که اندازه تبدیل فوریه به صورت زیر بدست می آید:

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

### Frequency Spectra •

طیف توان به عنوان مربع اندازه در نظر گرفته می شود، یعنی:

$$\begin{aligned} P(u, v) &= |F(u, v)|^2 \\ &= R^2(u, v) + I^2(u, v) \end{aligned}$$

برای مثال به شکل زیر توجه کنید:



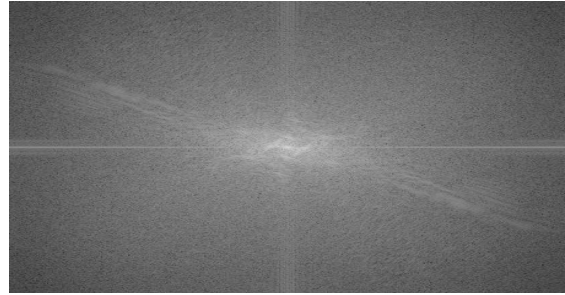
و زاویه فاز تبدیل برابر است :

$$\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$$

تبدیل فوریه پیوسته و معکوس آن (برگشت پذیر)

$$F(\mu) = \int_{-\infty}^{+\infty} f(t) e^{-2j\pi\mu t} dt$$

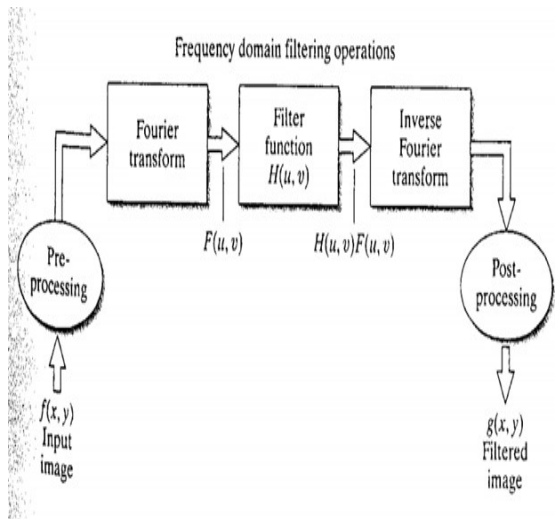
$$f(t) = \int_{-\infty}^{+\infty} F(\mu) e^{2j\pi\mu t} d\mu$$



طیف تصویر بالا

## • فیلتر در حوزه فرکانس و تبدیل فوریه:

برای فیلتر در حوزه فرکانس مانند مراحل شکل زیر باید عمل کرد.



فیلترینگ در چند مرحله انجام میشود:  
گام اول: تعیین  $p$  و  $q$  سایز تصویر  $f$  دوبرابر سایز تصویر فعلی

گام دوم: ساختن تصویر  $f$  با افزودن صفر به تصویر

گام سوم: ضرب کردن  $f$  در  $1 - x + y$  و انتقال مبدأ به مرکز

گام چهارم: به دست آوردن  $dft$  حاصل

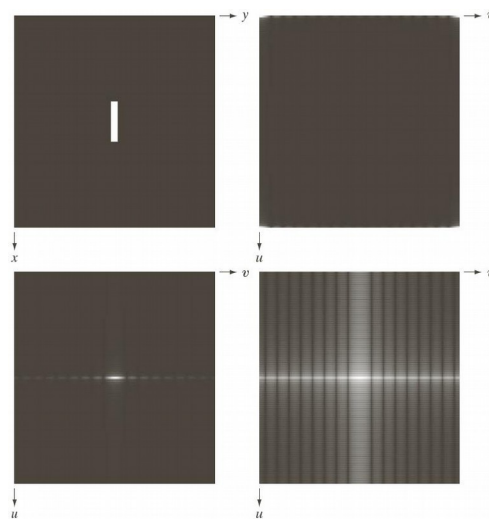
گام پنجم: اعمال فیلتر حقیقی در تصویر و ضرب آرایه ای آن در تصویر

گام ششم: به دست آوردن  $idft$  حاصل

گام هفتم: انتخاب بخش حقیقی و نتیجه را در  $1 - x + y$  ضرب کرده

## • Uncertainty Relation

این اصل بیان میکند که همواره باید حاصل ضرب تغییرات  $x$  و  $y$  در حاصل ضرب تغییرات  $u$  و  $v$  در حوزه فرکانس بزرگتر یا مساوی  $16\pi^2$  باشد و همینطور بیان میکند که یک شی کوچک در فضای مکان فرکانس بالایی خواهد داشت و بالعکس مانند تصویر زیر:



و طیف حوزه فرکانس عمود خواهد بود بر طیف اصلی

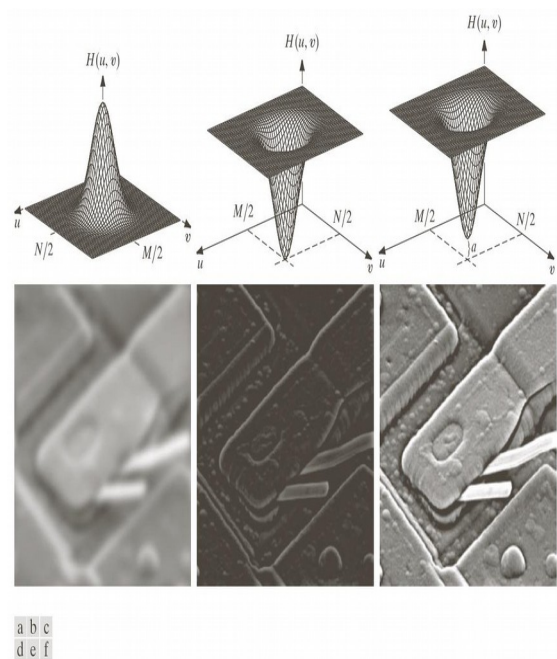


FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used  $a = 0.85$  in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).

به ترتیب از سمت چپ پایین گذر و بعدی بالاگذر  
انواع زیادی از این فیلترها وجود دارد از Low pass ها میتوان به گوسین و باتروث نام برد که سبب هموار سازی تصویر خواهند شد و همین طور میتوان از بالا گذر ها برای شارژ کردن و تیز کردن تصویر استفاده کرد

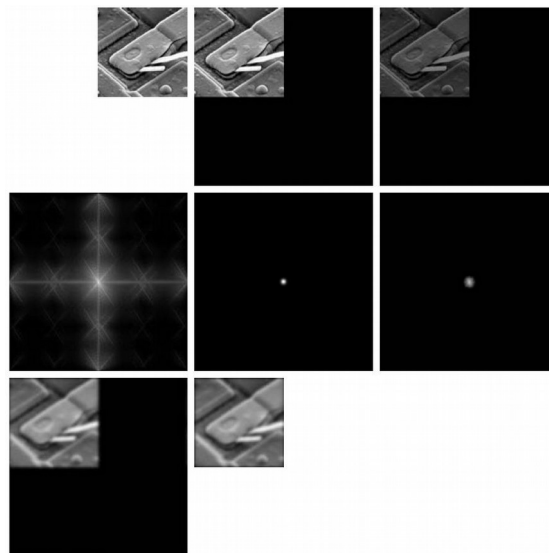


FIGURE 4.42 (a) Original image. (b)-(f) Results of filtering using ILPEs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

نمونه‌ای از هموار ساز است

گام هشتم: برداشتن بخشی از تصویر به اندازه  $n \times m$  از گوشه سمت چپ تصویر

همانند شکل زیر مراحل را:



### • Filters در حوزه فرکانس

در این حوزه فیلترهای متنوعی وجود دارد از جمله: بالاگذر پایین گذر و میان گذر.

#### فیلتر بالا گذر:

فیلتری است که [فرکانس‌های](#) بالاتر از مقدار خاصی را می‌گذراند و فرکانس‌های پایین‌تر از آن را عبور نمی‌دهد (تضعیف می‌کند). این نوع فیلترها به انواع متفاوتی ساخته می‌شوند

#### فیلتر پایین گذر:

نوعی از فیلتر می‌باشد که سیگنال‌هایی با [بسامدی](#) کمتر از بسامدی مشخص را عبور می‌دهد. [۱] لازم به توجه است که این فیلتر در [بردارش سیگنال](#) اهمیت دارد و اثری بر سیگنال [DC](#) ندارد.

#### فیلتر میان گذر:

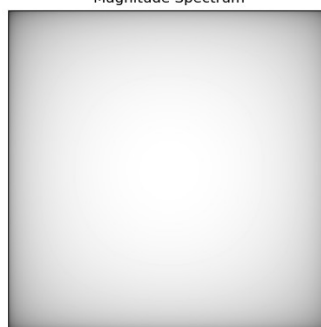
فیلتر میان گذر فیلتری است که محدوده ای خاص از فرکانس را از خود عبور می دهد

### 3-شرح نتایج:

در تمرین اول ابتدا ما فیلتر ها و تصویر را به حوزه فوریه برده و و پس از اعمال به حوزه مکان بر میگردانیم در ادامه طیف های فیلتر ها را نمایش میدهیم:

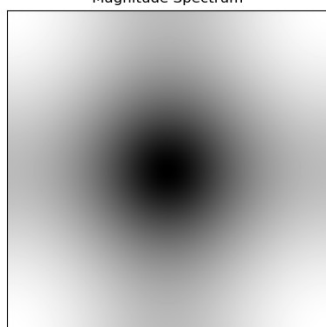
فیلتر اول:

Magnitude Spectrum



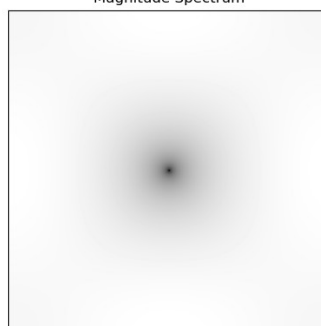
فیلتر سوم:

Magnitude Spectrum



فیلتر دوم:

Magnitude Spectrum



که به ترتیب فیلتر اول گوسین هموار ساز و دومی برای Edge detection و سومی برای Edge enhancement است فیلتر اول فقط جدا پذیر است به صورت زیر جدا میشود:

$$\frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{4} [1 \quad 2 \quad 1] = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

پس میتوان به صورت جمع دو عکس که جدا جدا هرکدام از فیلترها اعمال شده نوشت: که تصویر حاصل از اولین فیلتر:



وسپس تصویر نهایی که حاصل جمع هر دو  
است

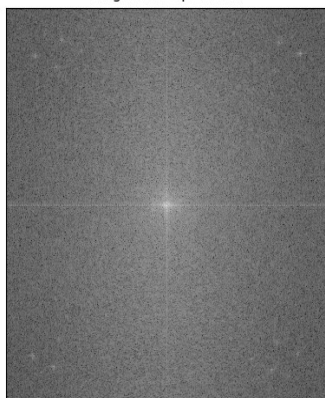


تصویر حاصل از دومین فیلتر:

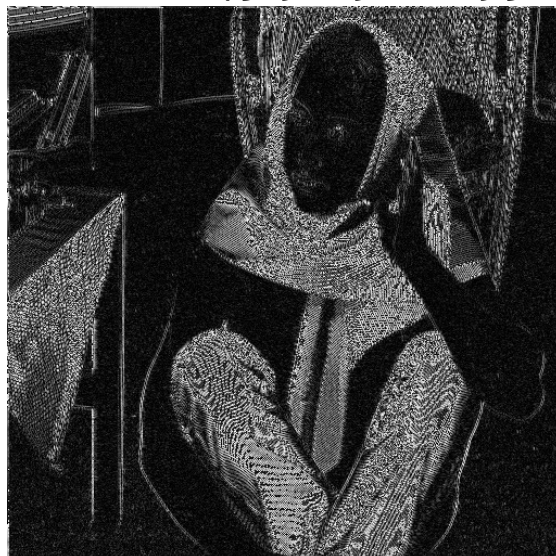
که smooth شده است



Magnitude Spectrum



تصویر حاصل از فیلتر دوم:



طیف بدون شیفیت برای تصویر lena

Magnitude Spectrum



و تصویر آخر:



تمرین دوم:

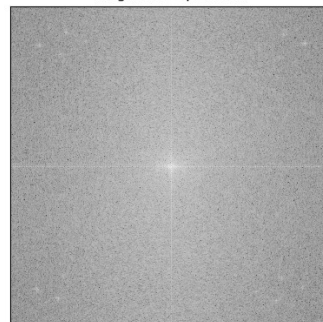
طیف با شیفیت برای تصویر lena

طیف با شیفیت و لاگ برای lena:

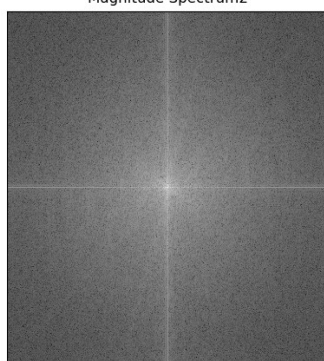
تصویر دوم

طیف با شیفت برای تصویر F16

Magnitude Spectrum

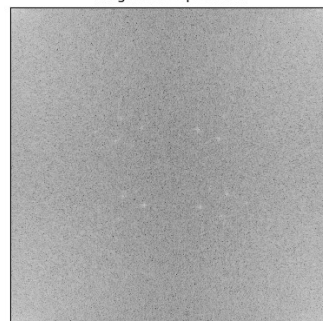


Magnitude Spectrum2



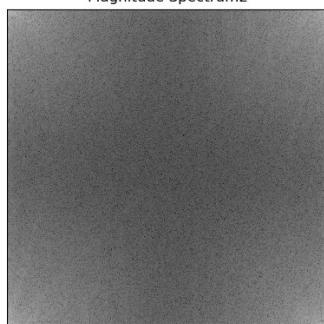
طیف با لاگ برای lena:

Magnitude Spectrum



طیف بدون شیفت برای تصویر F16

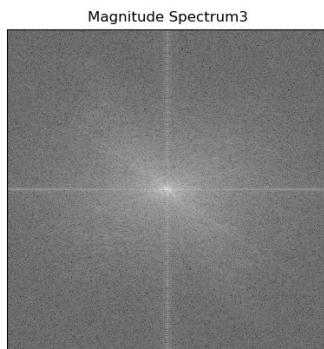
Magnitude Spectrum2



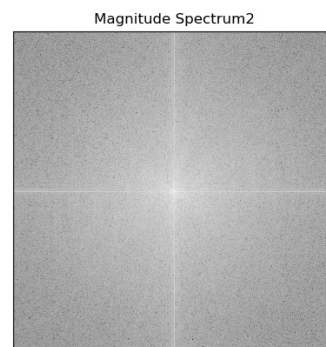


تصویر سوم:

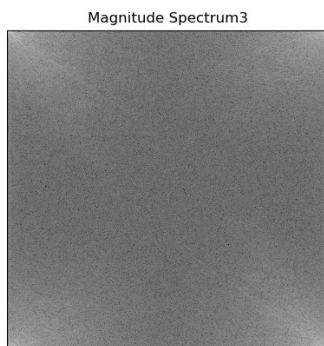
طیف با شیفت برای تصویر Baboon



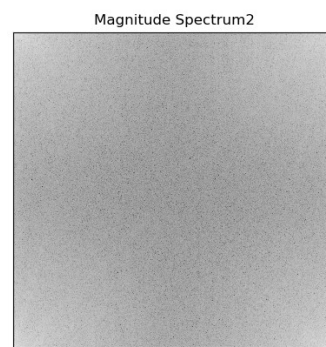
طیف با شیفت و لاگ برای تصویر F16



طیف بدون شیفت برای تصویر Baboon

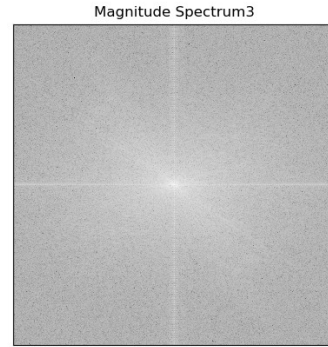
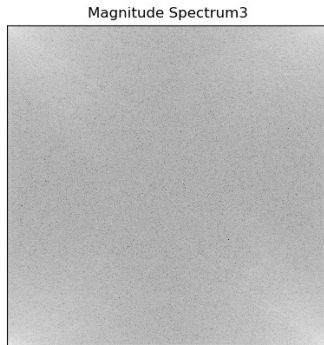


طیف با لاگ برای تصویر F16



طیف با شیفت و لاگ برای تصویر Baboon

طیف با لاگ برای تصویر Baboon



نتیجه اینکه از شیفت برای انتقال مبدأ به مرکز باعث میشه پراکندگی در مرکز باشد و لاگ هم باعث می‌شود که آن قسمت اصلی با بقیه جاها کمتر تفاوت داشته باشد و روشنتر خواهد شد.

،  $F(N-1, N-1)$  بالاترین فرکانس خواهد داشت. به طور کب زیرا هرچه از مرکز دورتر شویم فرکانس آن بالاتر است زیرا این توابع سنوسوید هستند و بیشترین فرکانس در قله آن هاست

سؤال 3:

خب میدانیم برای فیلترینگ ابتدا باید مراحل زیر را تکرار کنیم

گام اول: تعیین  $p$  و  $q$  سایز تصویر  $f$  دوبرابر سایز تصویر فعلی

گام دوم: ساختن تصویر  $f$  با افزودن صفر به تصویر

گام سوم: ضرب کردن  $f$  در  $1 - x + y$  و انتقال مبدأ به مرکز

گام چهارم: به دست آوردن  $dft$  حاصل



اصلی



اعمال فیلتر پایین گذر  $1/8$

گام پنجم: اعمال فیلتر حقیقی در تصویر و ضرب آرایه ای آن در تصویر

گام ششم: به دست آوردن  $\text{idft}$  حاصل

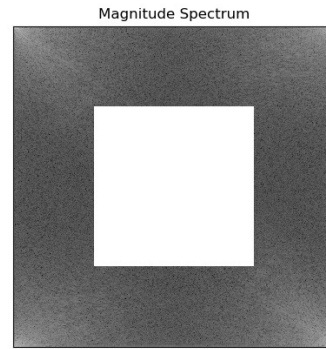
گام هفتم: انتخاب بخش حقیقی و نتیجه را در  $1 -$  به توان  $x+y$  ضرب کرده

گام هشتم: برداشتن بخشی از تصویر به اندازه  $n*m$  از گوشه سمت چپ تصویر

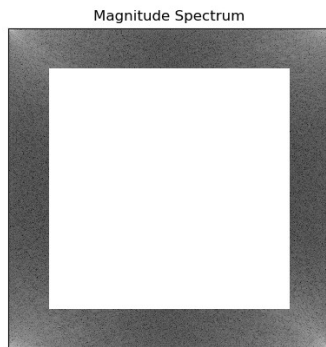
و سپس کلاً در حوزه فرکانس به دلیل اینکه فیلتر باید ضرب آرایه ای شود باید سائز فیلتر و عکس برابر باشند پس در اینجا باید ببایم و روی فیلتر مان هم همه مراحل قبل را بزنیم و سائز آن را با سائز تصویر برابر کرده و سپس با تصویر ضرب آرایه ای میکنیم

و اینکه کجاها باهم برابر میشن به پدینگ مربوط است .

سؤال 4:  
قسمت اول



طیف برای  $t=1/4$



و این هم طیف حاصل و دیده می‌شود که به

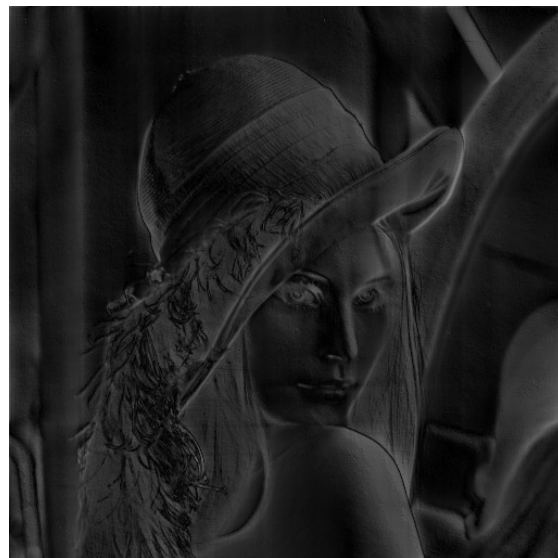
دلیل پایین گذر بودن فقط فرکانس های پایین را عبور میدهد همان طور که ملاحظه میکنید و تصویر هموارتر خواهد شد

طیف برای  $1/8$

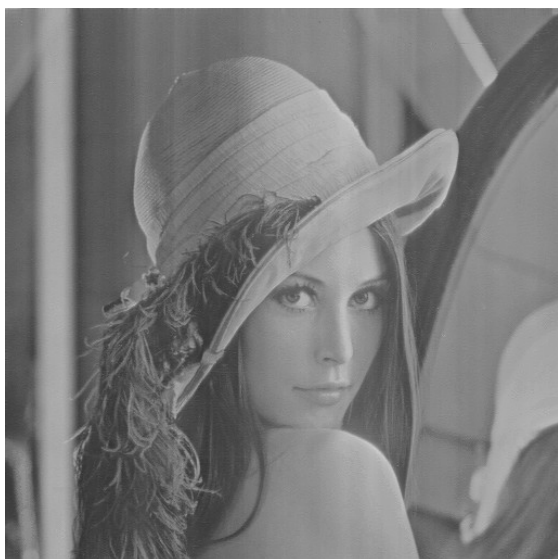


$t=1/4$

:b  
ابتدا برای تمسای 1/4  
(الف)

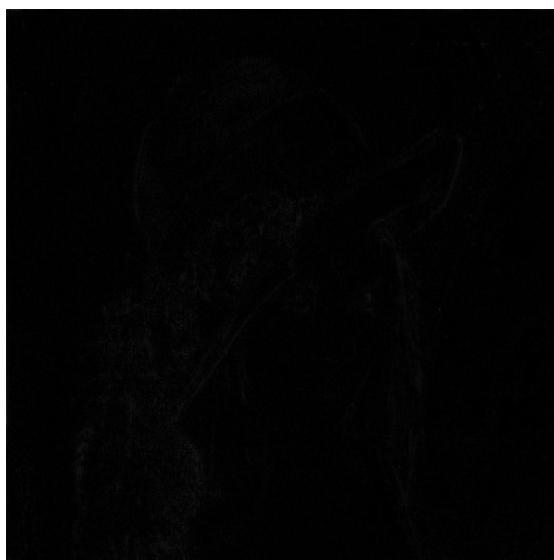


(ج)



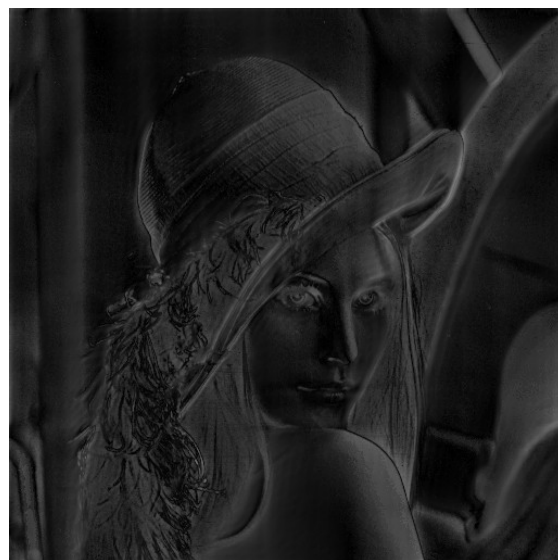
(ب)





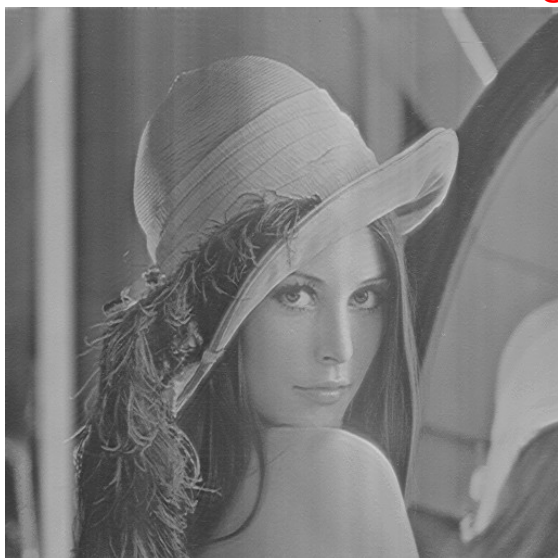
هرسه باهم:

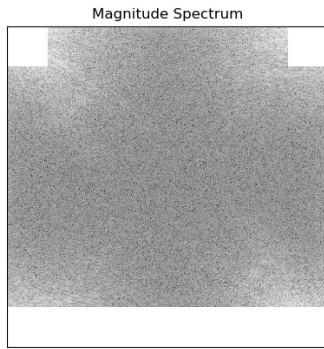
برای  $t=1/8$   
الف



(ب)

(ج)





هرسه باهم:

### طیف هر سه باهم

در این قسمت هر کدام از این‌ها در طیف مربوطه یک مربع در گوشه‌اش و یک مستطیل در پایین آن ایجاد میکنند و هر سه باهم یک فیلتر بالا گذر را تسکیل میدهند که فیلتر بالا گذر آن فرکانس های بالا را عبور داده و شکل همان چیزی که در بالا است میشود.



### 4-پیوست کد:

کد تمرین اول که ابتدا می‌آییم هم فیلتر و هم تصویر را به حوزه فوریه می‌بریم



Activities Visual Studio Code Sun 20:36 fa

fourie.py - Visual Studio Code

File Edit Selection View Go Debug Terminal Help

magnitude.py fourieakhar.py guassian.py fourie b.py fourie.py

```
home > f > Desktop > fourie.py > ...
1 import cv2 as cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4 image = cv2.imread("/home/f/Pictures/Wallpapers/4/Barbara.bmp")
5 image_c = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
6 filter1 = np.array([[1], [2], [1]])/4
7 filter2 = np.array([[1, 2, 1]])/4
8 new_image = np.zeros((1024,1024),np.float64)
9 new_filter=np.zeros((1024,1024),np.float64)
10 new_filter2=np.zeros((1024,1024),np.float64)
11 for a in range(0,512):
12     for b in range(0,512):
13         new_image[a][b] = image_c[a][b]
14
15
16
17 new = np.fft.fft2(new_image)
18 fourier_img = np.fft.fftshift(new)
19
20 for a in range(0,3):
21     for b in range(0,1):
22         new_filter[a][b] = filter1[a][b]
23
24
25 fshift= np.fft.fft2(new_filter)
26 mask = np.fft.fftshift(fshift)
27
28
29 for a in range(0,1):
30     for b in range(0,3):
31         new_filter2[a][b] = filter2[a][b]
32
```

Python 3.7.4 64-bit ('base': conda) 25 1 Ln 78, Col 10 Spaces: 4 UTF-8 LF Python 2

Activities Visual Studio Code Sun 20:38 en

fourie.py - Visual Studio Code

File Edit Selection View Go Debug Terminal Help

magnitude.py fourieakhar.py guassian.py fourie b.py fourie.py

```
home > f > Desktop > fourie.py > ...
28
29 for a in range(0,1):
30     for b in range(0,3):
31         new_filter2[a][b] = filter2[a][b]
32
33
34 fshift2= np.fft.fft2(new_filter2)
35 mask2 = np.fft.fftshift(fshift2)
36 '''
37 magnitude_spectrum1 = 20*np.log(np.abs(mask))
38
39 plt.subplot(),plt.imshow(magnitude_spectrum1, cmap = 'gray')
40 plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
41 plt.show()'''
42
43 #mask = np.resize(mask,(1024,1024))
44
45
46 # multiply resize filter and image
47 final_image = [[0 for x in range(1024)] for y in range(1024)]
48 final_image2 = [[0 for x in range(1024)] for y in range(1024)]
49
50 for i in range(1024):
51     for j in range(1024):
52         final_image[j][i] = fourier_img[j][i] * mask[j][i]
53 final = np.fft.ifftshift(final_image)
54
55
56 for i in range(1024):
57     for j in range(1024):
58         final_image2[j][i] = fourier_img[j][i] * mask2[j][i]
59 final2 = np.fft.ifftshift(final_image2)
```

Python 3.7.4 64-bit ('base': conda) 25 1 Ln 78, Col 10 Spaces: 4 UTF-8 LF Python 2

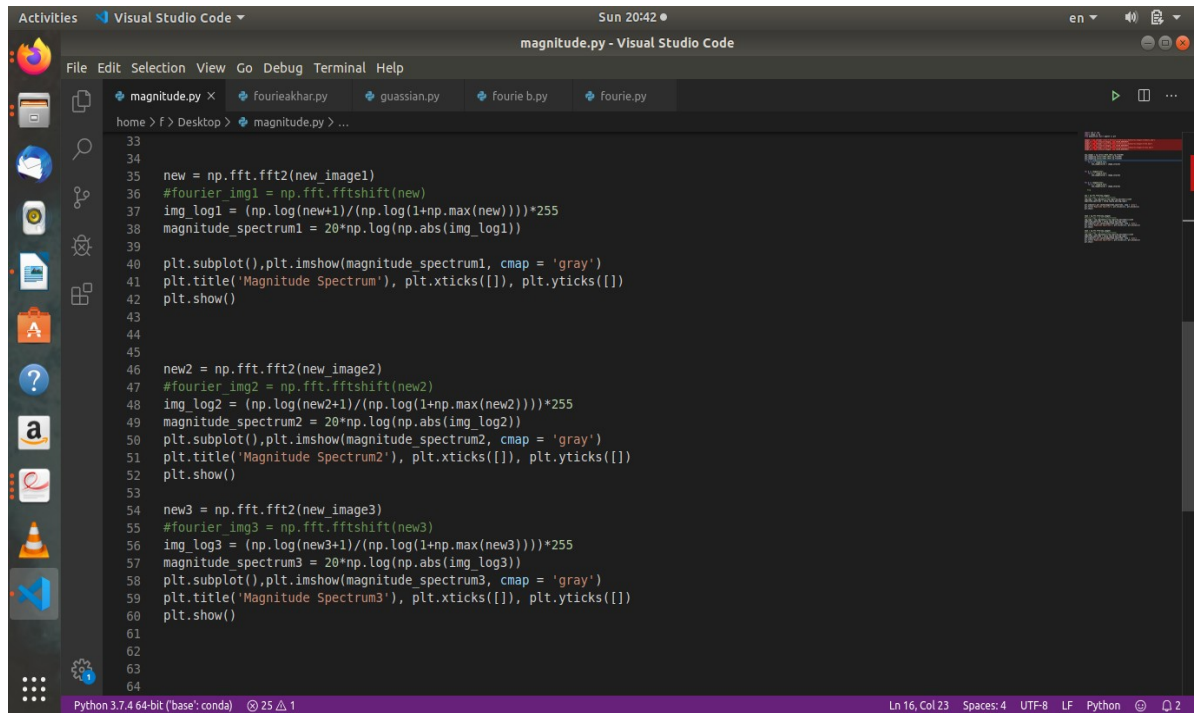
```
home > f > Desktop > fourie.py > ...
43 #mask = np.resize(mask,(1024,1024))
44 # multiply resize filter and image
45 final_image = [[0 for x in range(1024)] for y in range(1024)]
46 final_image2 = [[0 for x in range(1024)] for y in range(1024)]
47
48 for i in range(1024):
49     for j in range(1024):
50         final_image[j][i] = fourier_img[j][i] * mask[j][i]
51 final = np.fft.ifftshift(final_image)
52
53
54 for i in range(1024):
55     for j in range(1024):
56         final_image2[j][i] = fourier_img[j][i] * mask2[j][i]
57 final2 = np.fft.ifftshift(final_image2)
58 ff = [[0 for x in range(1024)] for y in range(1024)]
59 ff2 = [[0 for x in range(1024)] for y in range(1024)]
60 '''ss = [[0 for x in range(1024)] for y in range(1024)]'''
61 ff = np.fft.ifft2(final)
62 ff2 = np.fft.ifft2(final2)
63 f = np.zeros((1024, 1024),np.uint8)
64 f2 = np.zeros((1024, 1024),np.uint8)
65 for n in range(1024):
66     for m in range(1024):
67         f[n][m] = np.abs(ff[n][m])
68 for n in range(1024):
69     for m in range(1024):
70         f2[n][m] = np.abs(ff2[n][m])
71 ss=f+f2/2
72 ss=ss[0:512,0:512]
73 cv2.imshow("final.png",ss)
74 cv2.waitKey(0)
```

که  
از  
fft  
ifft  
و  
fftsh  
ift

برای شیف و به دست آوردن فوریه  
استفاده میکنیم  
تمرین دوم:

```
home > f > Desktop > magnitude.py > ...
1 import cv2 as cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 image1 = cv2.imread("/home/f/Desktop/term6/binai/Homeworks/Images/4/Baboon.bmp")
6 image_c1 = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)
7 image2 = cv2.imread("/home/f/Desktop/term6/binai/Homeworks/Images/4/F16.bmp")
8 image_c2 = cv2.cvtColor(image2, cv2.COLOR_BGR2GRAY)
9 image3 = cv2.imread("/home/f/Desktop/term6/binai/Homeworks/Images/4/Lena.bmp")
10 image_c3 = cv2.cvtColor(image3, cv2.COLOR_BGR2GRAY)
11
12
13 new_image1 = np.zeros((1024,1024),np.float64)
14 new_image2=np.zeros((1024,1024),np.float64)
15 new_image3=np.zeros((1024,1024),np.float64)
16 for a in range(0,512):
17     for b in range(0,512):
18         new_image1[a][b] = image_c1[a][b]
19
20
21
22 for a in range(0,512):
23     for b in range(0,512):
24         new_image2[a][b] = image_c2[a][b]
25
26
27
28 for a in range(0,512):
29     for b in range(0,512):
30         new_image3[a][b] = image_c3[a][b]
31
32 #log
```

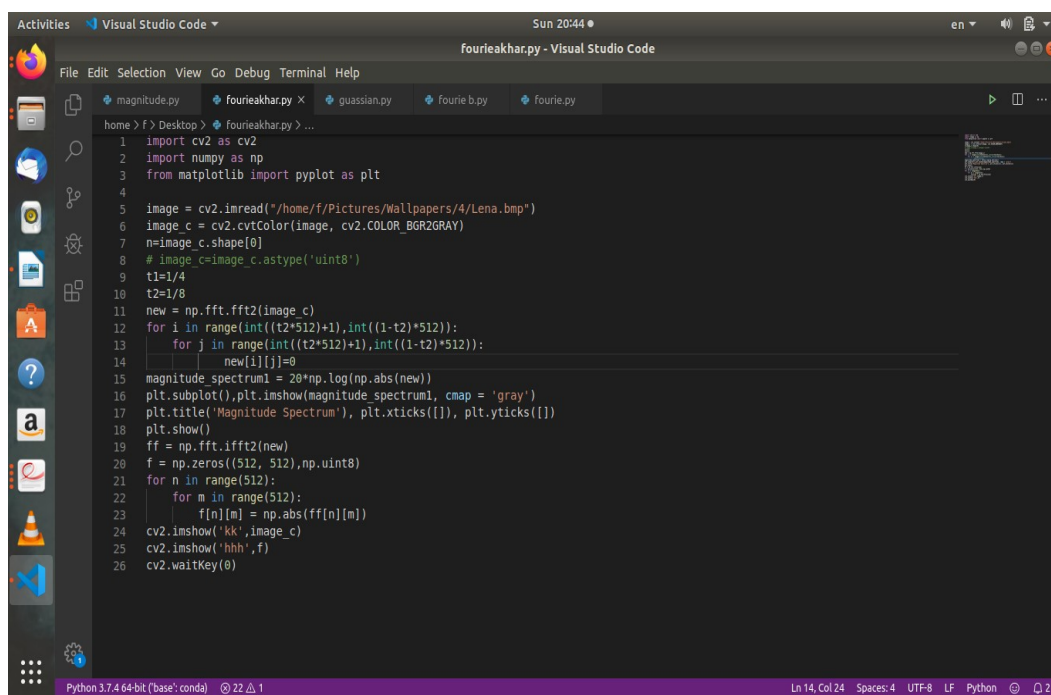
تمرین 2 ادامه:  
به دست آوردن طیف:



```
33
34
35 new = np.fft.fft2(new_image1)
36 #fourier_img1 = np.fft.fftshift(new)
37 img_log1 = (np.log(new+1)/(np.log(1+np.max(new))))*255
38 magnitude_spectrum1 = 20*np.log(np.abs(img_log1))
39
40 plt.subplot(),plt.imshow(magnitude_spectrum1, cmap = 'gray')
41 plt.title('Magnitude Spectrum'), plt.xticks([], plt.yticks([]))
42 plt.show()
43
44
45
46 new2 = np.fft.fft2(new_image2)
47 #fourier_img2 = np.fft.fftshift(new2)
48 img_log2 = (np.log(new2+1)/(np.log(1+np.max(new2))))*255
49 magnitude_spectrum2 = 20*np.log(np.abs(img_log2))
50 plt.subplot(),plt.imshow(magnitude_spectrum2, cmap = 'gray')
51 plt.title('Magnitude Spectrum2'), plt.xticks([], plt.yticks([]))
52 plt.show()
53
54 new3 = np.fft.fft2(new_image3)
55 #fourier_img3 = np.fft.fftshift(new3)
56 img_log3 = (np.log(new3+1)/(np.log(1+np.max(new3))))*255
57 magnitude_spectrum3 = 20*np.log(np.abs(img_log3))
58 plt.subplot(),plt.imshow(magnitude_spectrum3, cmap = 'gray')
59 plt.title('Magnitude Spectrum3'), plt.xticks([], plt.yticks([]))
60 plt.show()
61
62
63
64
```

تمرین آخر قسمت اول:

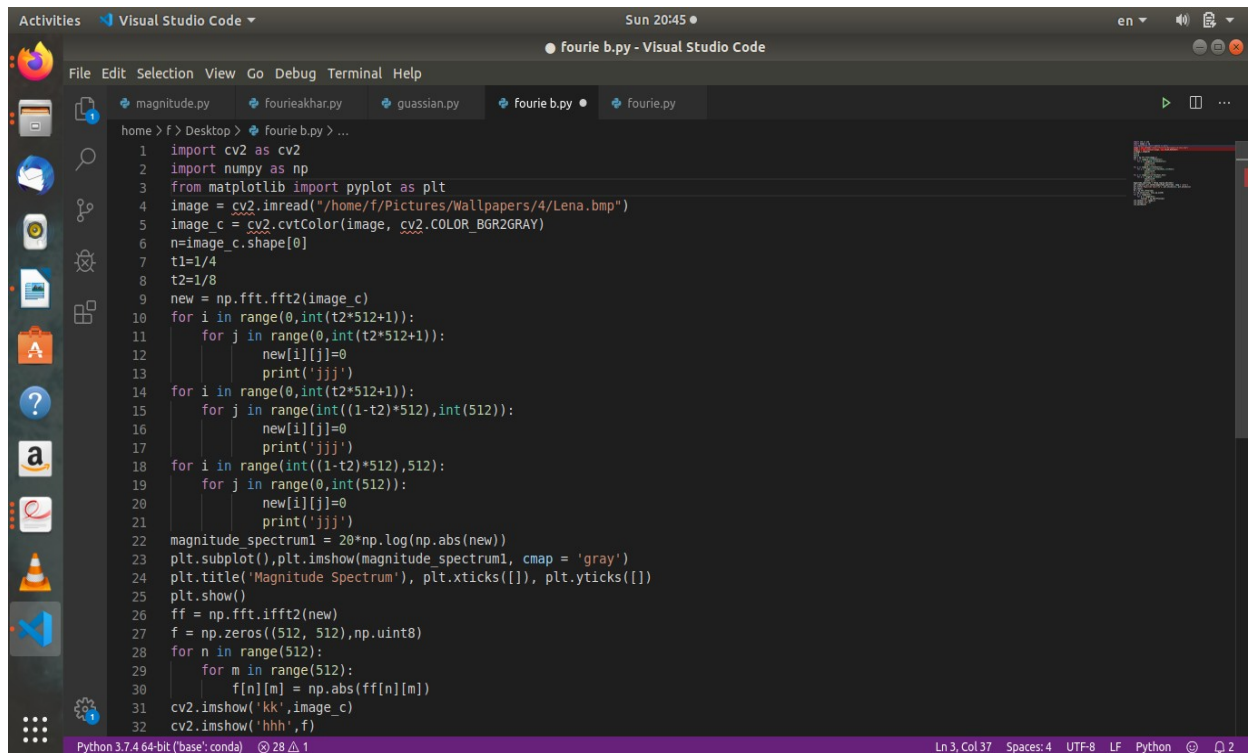
چک کردن برقراری شرط و صفر کردن  
پیکسلها



```
1 import cv2 as cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 image = cv2.imread("/home/f/Pictures/Wallpapers/4/Lena.bmp")
6 image_c = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7 n=image_c.shape[0]
8 # image_c=image_c.astype('uint8')
9 t1=1/4
10 t2=1/8
11 new = np.fft.fft2(image_c)
12 for i in range(int((t2*512)+1),int((1-t2)*512)):
13     for j in range(int((t2*512)+1),int((1-t2)*512)):
14         new[i][j]=0
15 magnitude_spectrum1 = 20*np.log(np.abs(new))
16 plt.subplot(1),plt.imshow(magnitude_spectrum1, cmap = 'gray')
17 plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
18 plt.show()
19 ff = np.fft.ifft2(new)
20 f = np.zeros((512, 512),np.uint8)
21 for n in range(512):
22     for m in range(512):
23         f[n][m] = np.abs(ff[n][m])
24 cv2.imshow('kk',image_c)
25 cv2.imshow('hhh',f)
26 cv2.waitKey(0)
```

Python 3.7.4 64-bit (base: conda) Ln 14, Col 24 Spaces: 4 UTF-8 LF Python

تمرین آخر قسمت دوم همانند قسمت قبل  
و جدا جدا چک کردن شرط ها



```
1 import cv2 as cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4 image = cv2.imread("/home/f/Pictures/Wallpapers/4/Lena.bmp")
5 image_c = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
6 n=image_c.shape[0]
7 t1=1/4
8 t2=1/8
9 new = np.fft.fft2(image_c)
10 for i in range(0,int(t2*512+1)):
11     for j in range(0,int(t2*512+1)):
12         new[i][j]=0
13         print('jjj')
14 for i in range(0,int(t2*512+1)):
15     for j in range(int((1-t2)*512),int(512)):
16         new[i][j]=0
17         print('jjj')
18 for i in range(int((1-t2)*512),512):
19     for j in range(0,int(512)):
20         new[i][j]=0
21         print('jjj')
22 magnitude_spectrum1 = 20*np.log(np.abs(new))
23 plt.subplot(1,2,1),plt.imshow(magnitude_spectrum1, cmap = 'gray')
24 plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
25 plt.show()
26 ff = np.fft.ifft2(new)
27 f = np.zeros((512, 512),np.uint8)
28 for n in range(512):
29     for m in range(512):
30         f[n][m] = np.abs(ff[n][m])
31 cv2.imshow('kk',image_c)
32 cv2.imshow('hhh',f)
```



برای نوشتن روابط ریاضی ابزار Equation Editor از کارایی بسیار بالایی برخوردار است. تمامی نمادهای مورد نیاز در این ابزار پیش بینی شده است. توضیحات تمام متغیرها، پارامترها و نمادهای جدید در روابط، چنانچه پیش از آن توضیح داده نشده‌اند، باید بدون فاصله بعد از رابطه بیان شوند. مانند:

$$\frac{f'_{cc}}{f'_c} = \left[ 1 + a \left( \frac{f_l}{f'_c} \right) \right] \quad (1)$$

$$\frac{\varepsilon'_{cc}}{\varepsilon_c} = \left[ 1 + b \left( \frac{f_l}{f'_c} \right) \right] \quad (2)$$

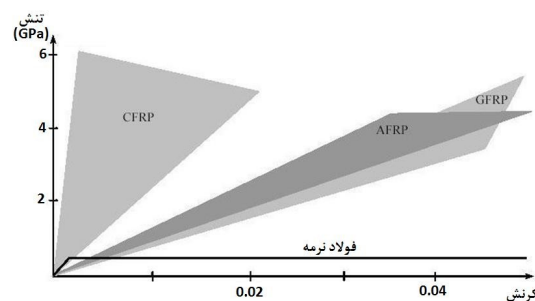
که در آن  $f'_c$  و  $\varepsilon'_c$ ، به ترتیب مقاومت فشاری و کرنش محوری مربوط به تنش حداکثر بتن محصور نشده می‌باشند. همچنین  $f_l$ ،  $\varepsilon'_l$  فشار محصور شدگی بوده و مقادیر  $a$  و  $b$  ضرایبی هستند که به صورت تجربی تعیین شده‌اند.

#### 4- نتیجه گیری

نتایج اصلی و مهم تحقیق، در این بخش ذکر می‌شود. در این مقاله نمونه، مشخصات یک مقاله آماده به چاپ برای مجله علمی و پژوهشی مدل سازی در مهندسی بیان گردید. قسمت‌های مهمی که می‌بایست مورد دقت کافی قرار گیرند، مواردی نظیر ابعاد و حواشی صفحه و ستونها، نحوه تهیه عنوان و چکیده به فارسی و انگلیسی، بخشهای اصلی، نحوه شماره گذاری بخشها و زیربخشها، نحوه شماره گذاری جدولها، شکلها و روابط ریاضی و ارجاع به آنها، فهرستبندی، مرتب سازی و شماره گذاری مراجع، و بالاخره اندازه و نوع قلمها را شامل میشوند.

چنانچه شکل یا جدولی در یک ستون جا نگیرد، میتواند آن را در دو ستون قرار داد. سعی شود شکل از نرم افزار مستقیماً به محیط Word آورده شده و با استفاده از ابزار Edit Picture بازبینی و ویرایش شود. دقت شود که ویژگی Layout در شکل به صورت In line with text و در یک سطر مستقل درج شده باشد تا در صفحه بندی مشکلی به وجود نیاید. در هر حال، شکلها باید به صورتی واضح و با توضیحات کافی و داخل همان کادر متن در مقاله درج شوند و با سطرهای قبل و بعد فاصله کافی داشته باشند. هرگز از اسکن کردن شکل چاپ شده استفاده نشود. سعی شود برای وضوح هرچه بیشتر شکلها، با بزرگ کردن شکل از تمام عرض ستون استفاده شود.

عدم رعایت قواعد برچسبگذاری و واحد نویسی محورها در نمودارها اغلب موجب کاهش رسایی مقاله میشود. برای کلیه محورها به جای استفاده از حروف و نمادها از کلمات استفاده شده و واحد هر یک داخل پرانتز قرار داده شود. اگر از شکلهای مراجع دیگری در مقاله خود استفاده میکنید، ضروری است که نام و نشانی منبع در زیر شکل ذکر شود. شکل 1 نمونه‌ای از چگونگی تنظیم یک شکل را نمایش می‌دهد.



شکل 1- منحنی‌های تنش-کرنش کامپوزیت‌های FRP در مقایسه با فولاد نرمه [1]

#### مراجع

[1] J. Author1, B. Author2, and K. Author3, "Title of Paper", IEEE Transactions on Energy Conversion, Vol. x, NO. xx, August 2002, pp. 363 – 367.



[2] G. Author1, K. Author2, and A. Author3 "Title of Paper", Name of Conference, Sydney, Australia, Vol. x, August 2002, pp. 338-343.

[3] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, Linear Programming and Network Flows, 4<sup>th</sup> ed., Wiley, NJ, USA, 2010.

[4] اصغر زارع و علی محمدزاده، "حذف نویز ضربه‌ای از تصاویر دیجیتالی مبتنی بر تخمین توزیع مکانی نویزها"، نشریه مدل‌سازی در مهندسی، دوره 12، شماره 39، زمستان 1393، صفحه 13-29.

[5] نام نویسنده اول، نام نویسنده دوم و نام نویسنده سوم، "عنوان مقاله"، نام کنفرانس، تهران، ایران، 7 تا 10 اردیبهشت، دوره 5، 1395.

[6] قدرت اله حیدری، طراحی الکتریکی خطوط انتقال نیرو، چاپ اول، انتشارات تابش برق، ایران، 1379.