

**دانشگاه صنعتی امیرکبیر**

**دانشکده مهندسی پزشکی**

**گزارش تمرین چهارم پردازش تصویر**

**دانشجو**

**سیدابوالفضل مرتضوی 9833063**

آذرماه 1401

فهرست مطالب

[تمرین اول) 3](#_Toc122165355)

[تمرین دوم) 4](#_Toc122165356)

[تمرین سوم) 4](#_Toc122165357)

[3‏-‏1 بخش1) 4](#_Toc122165358)

[3‏-‏2 بخش2) 4](#_Toc122165359)

[3‏-‏3 بخش3) 5](#_Toc122165360)

[3‏-‏4 بخش4) 7](#_Toc122165361)

[3‏-‏5 بخش‌های 5تا10 8](#_Toc122165362)

[سوال چهارم) 11](#_Toc122165363)

[4-‏1 بخش 1) 11](#_Toc122165364)

[4‏-‏2 بخش2) 11](#_Toc122165365)

[4‏-‏3 بخش 3) 12](#_Toc122165366)

[4‏-‏4 بخش4) 13](#_Toc122165367)

فهرست مطالب

[شکل ‏3.1 تصاویر خروجی بخش سوم 7](#_Toc122165381)

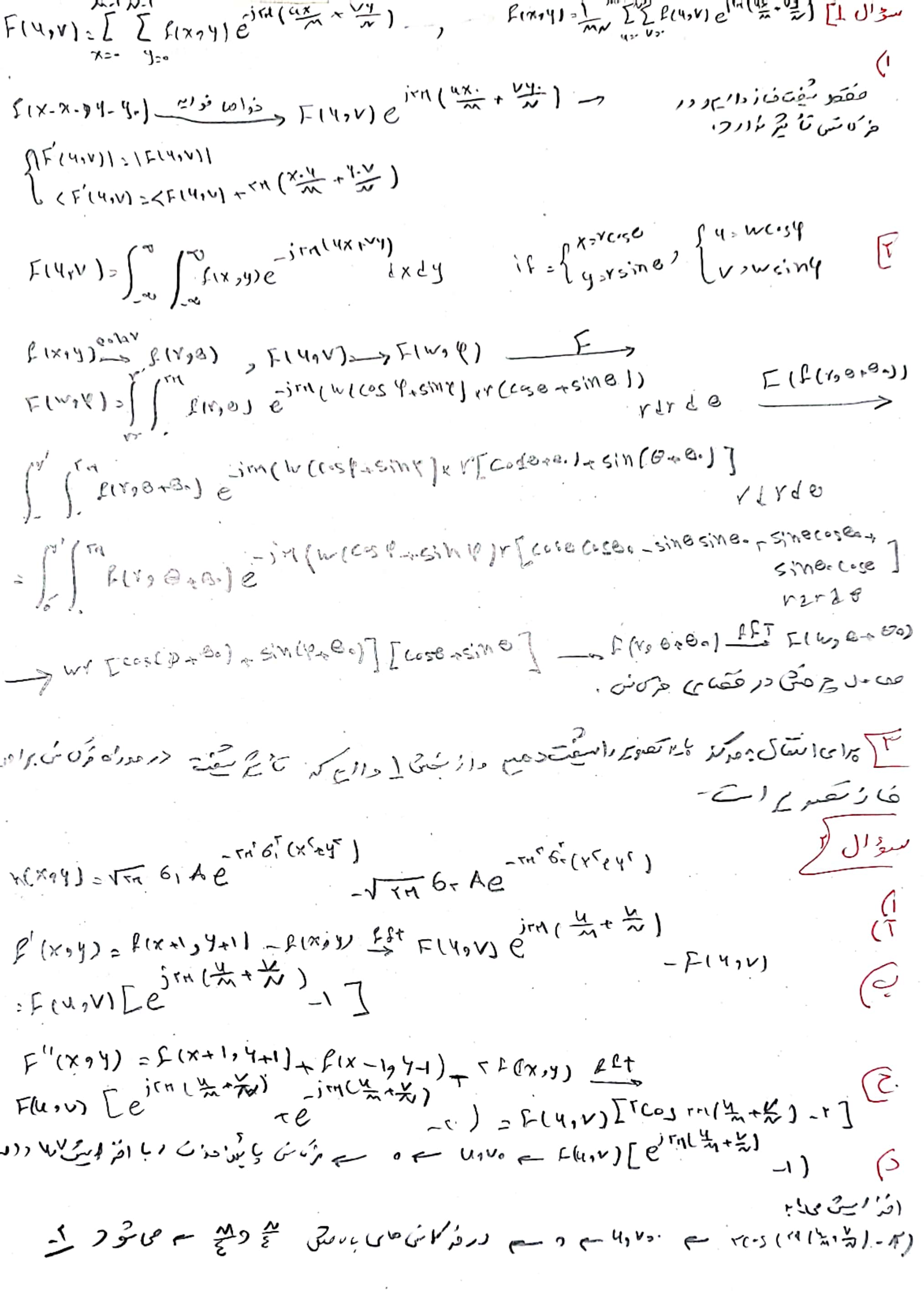
[شکل ‏3.2 خروجی بخش د 8](#_Toc122165382)

[شکل ‏3.3 خروجی نهایی 10](#_Toc122165383)

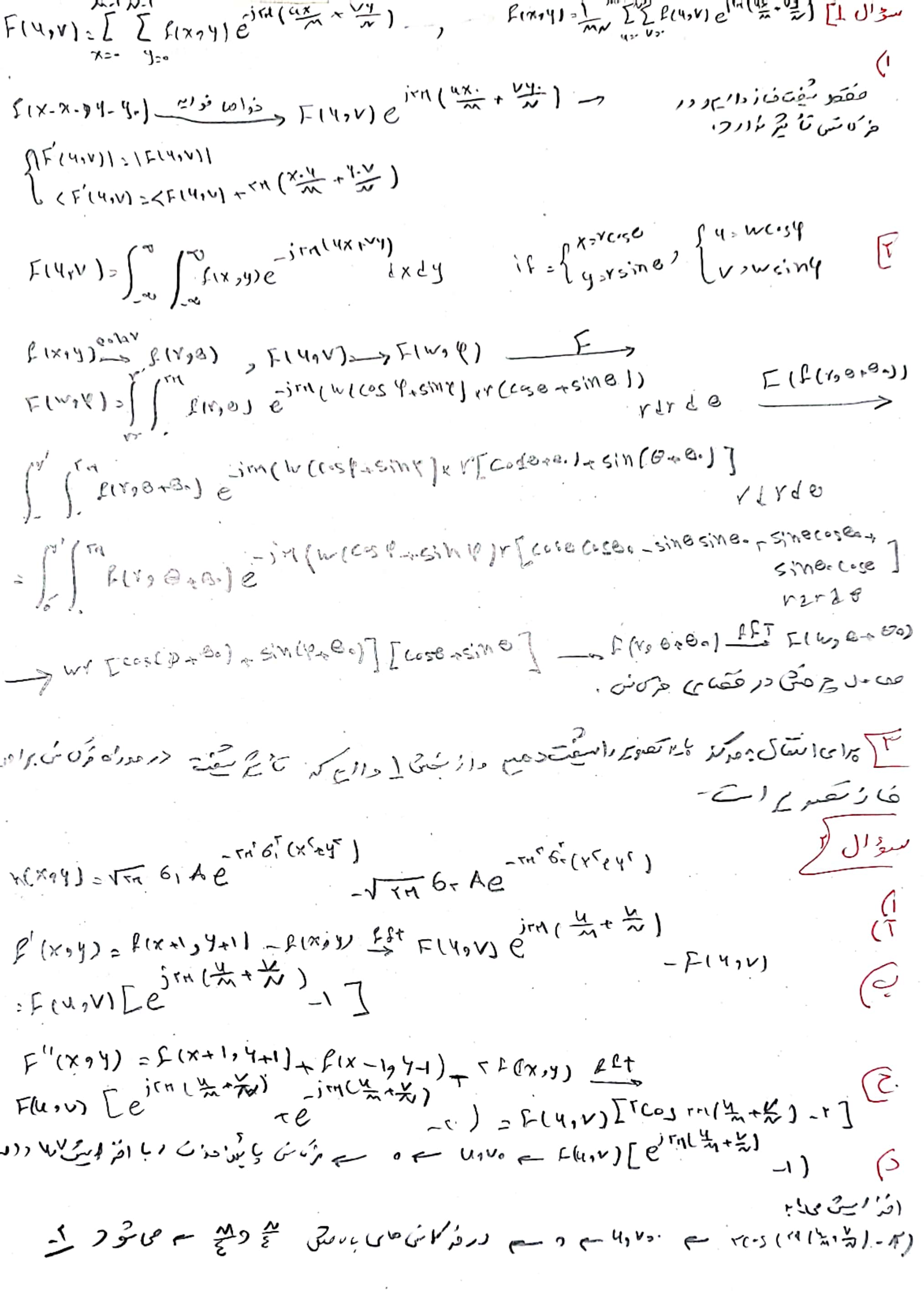
[شکل ‏4.1 تصاویر خروجی این بخش 12](#_Toc122165384)

[شکل ‏4.2 خروجی این بخش 14](#_Toc122165385)

# تمرین اول)



# تمرین دوم)



# تمرین سوم)

## بخش1)

created\_img=np.zeros([200,200]).astype('uint8')

created\_img[80:120,60:140]=255

با کد بالا این کار انجام شد.

## بخش2)

tranlate\_matrix\_x=np.float32(([1,0,20],[0,1,0]))

translated\_img\_x=cv.warpAffine(created\_img,tranlate\_matrix\_x,(200,200))

tranlate\_matrix\_y=np.float32(([1,0,0],[0,1,-40]))

translated\_img\_y=cv.warpAffine(created\_img,tranlate\_matrix\_y,(200,200))

rotate\_matrix\_30=cv.getRotationMatrix2D((100,100), 30, 1.0)

rotated\_img\_30=cv.warpAffine(created\_img, rotate\_matrix\_30, (200,200))

rotate\_matrix\_90=cv.getRotationMatrix2D((100,100), 90, 1.0)

rotated\_img\_90=cv.warpAffine(created\_img, rotate\_matrix\_90, (200,200))

با استفاده از کد بالا تصاویر خواسته‌شده ایجاد شدند.

## بخش3)

real\_img\_dft=np.fft.fft2(created\_img)

real\_img\_dft=np.fft.fftshift(real\_img\_dft)

real\_img\_phase=np.angle(real\_img\_dft)

trx\_img\_dft=np.fft.fft2(translated\_img\_x)

trx\_img\_dft=np.fft.fftshift(trx\_img\_dft)

trx\_img\_phase=np.angle(trx\_img\_dft)

try\_img\_dft=np.fft.fft2(translated\_img\_y)

try\_img\_dft=np.fft.fftshift(try\_img\_dft)

try\_img\_phase=np.angle(try\_img\_dft)

rot30\_img\_dft=np.fft.fft2(rotated\_img\_30)

rot30\_img\_dft=np.fft.fftshift(rot30\_img\_dft)

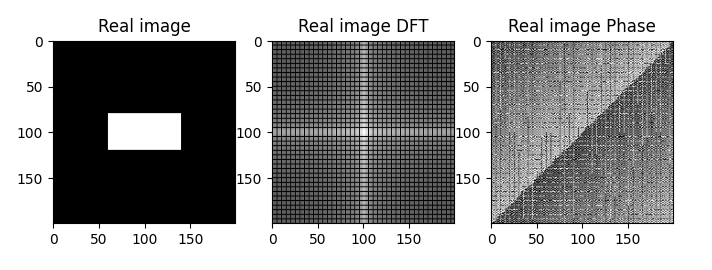
rot30\_img\_phase=np.angle(rot30\_img\_dft)

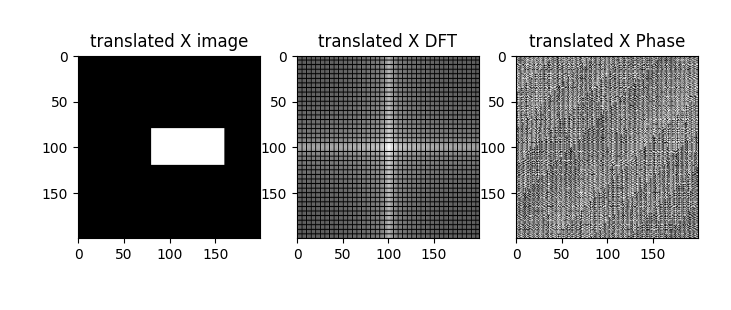
rot90\_img\_dft=np.fft.fft2(rotated\_img\_90)

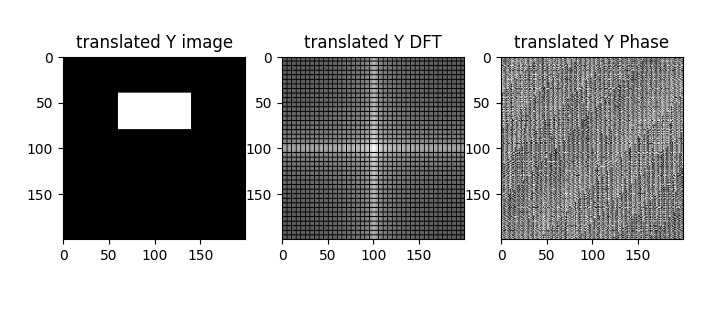
rot90\_img\_dft=np.fft.fftshift(rot90\_img\_dft)

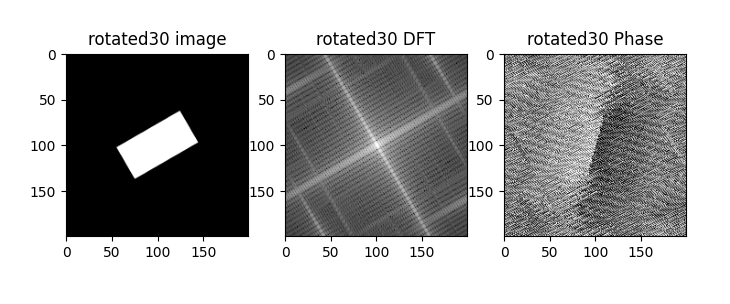
rot90\_img\_phase=np.angle(rot90\_img\_dft)

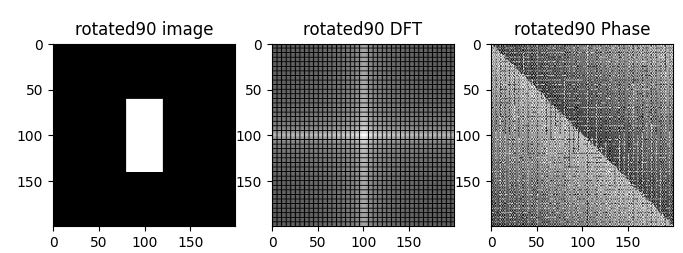
بااستفاده از کد بالا تبدیلات خواسته‌شده انجام شدند و خروجی درشکل ‏3.1 نشان داده شده است.











شکل ‏3.1 تصاویر خروجی بخش سوم

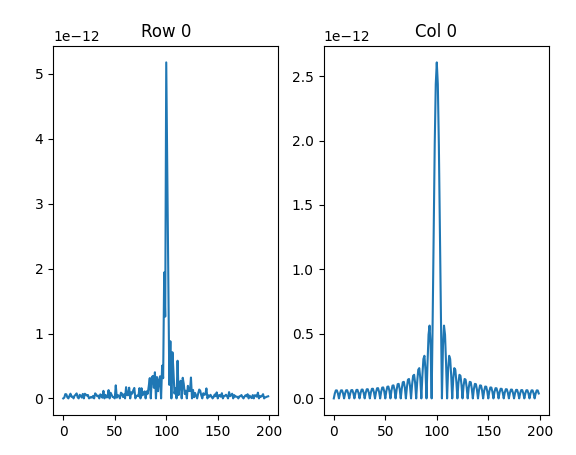
همانطور که در تصاویر بالا مشخص است حرکت درراستای X,Y تغییری در فضای فرکانسی ایجاد نمی‌کند اما در فاز باعث تغییرات می‌شود که این تغییرات به خاطر ضرب شدن ejx است. چرخش تصویر باعث تغییر در فضای فرکانسی و فضای فاز می‌شود. در چرخش 90 درجه به علت تقارن شکل فرکانسی تغییری در شکل دیده‌نمی‌شود.

## بخش4)

real\_img\_row0=np.abs(real\_img\_dft[0])

real\_img\_col0=np.abs(real\_img\_dft[:,0])

بااستفاده از کد بالا این کار انجام شد و خروجی درشکل ‏3.2 نشان داده شده است.



شکل ‏3.2 خروجی بخش د

## بخش‌های 5تا10

meta\_img=cv.imread('metacarpal.png',0)

meta\_fft=np.fft.fft2(meta\_img)

meta\_fft=np.fft.fftshift(meta\_fft)

meta\_fft\_max=abs(meta\_fft[0:200,0:200])

meta\_fft\_max=list(meta\_fft\_max)

max\_val=0

max\_cor=(0,0)

for i in range(200):

    if max(meta\_fft\_max[i])>max\_val:

        max\_val=max(meta\_fft\_max[i])

        max\_cor=(i,list(meta\_fft\_max[i]).index(max(meta\_fft\_max[i])))

cor\_x,cor\_y=max\_cor

meta\_fft[cor\_x,cor\_y]=0

meta\_fft[642-cor\_x,cor\_y]=0

reformed\_img=np.fft.ifftshift(meta\_fft)

reformed\_img=abs(np.fft.ifft2(reformed\_img))

R\_non\_rot=reformed\_img[318:338,230:250]

R\_non\_rot\_dft=np.fft.fft2(R\_non\_rot)

R\_non\_rot\_dft=np.fft.fftshift(R\_non\_rot\_dft)

R\_rot\_dft=R\_non\_rot\_dft[20::-1,20::-1]

R\_rot=np.fft.ifftshift(R\_rot\_dft)

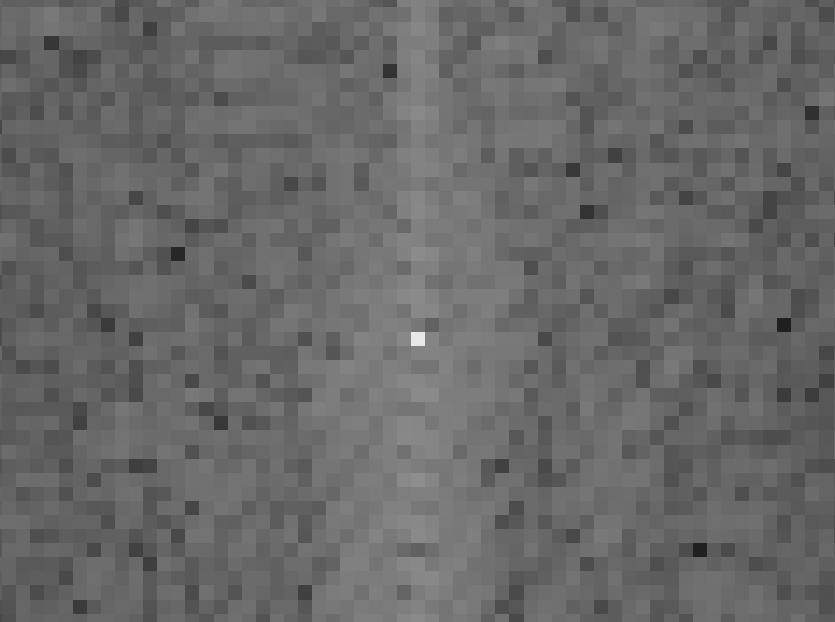
R\_rot=abs(np.fft.ifft2(R\_rot))

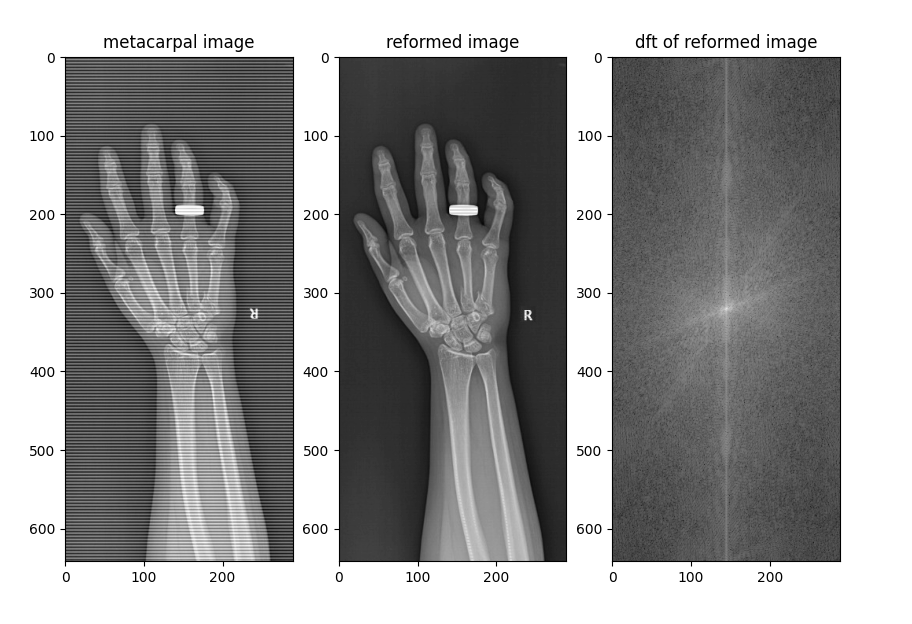
reformed\_img[318:338,230:250]=R\_rot

reformed\_img\_dft=np.fft.fft2(reformed\_img)

reformed\_img\_dft=np.log1p(np.abs(np.fft.fftshift(reformed\_img\_dft)))

بااستفاده از کد بالا کارهای خواسته شده انجام شد. برای حذف نویز دو نقطه سفید در مختصات (160,145) و (482,145) وجود داشتند که صفر شدند و در تصویر خطوط حذف شدند. خروجی نهایی درشکل ‏3.3 نشان داده شده است.





شکل ‏3.3 خروجی نهایی

# سوال چهارم)

## بخش 1)

shoulder\_img=cv.imread('shoulder.jpg',0)

## بخش2)

shoulder\_img=cv.imread('shoulder.jpg',0)

zero\_pad\_matrix=np.zeros([2048,2048])

zero\_pad\_matrix[0:1024,0:1024]=shoulder\_img

zero\_pad\_shoulder=zero\_pad\_matrix

centered\_shoulder=center\_img(zero\_pad\_shoulder)

shoulder\_img\_dft=np.fft.fft2(centered\_shoulder)

در این بخش ابتدا تصویر به اندازه 1024پیکسل از هرطرف پد شد و سپس با تابع center\_img تصویر برای انتقال به مرکز در حوزه فرکانس آماده شد.

def center\_img(img):

    centered\_img=np.zeros([img.shape[0],img.shape[1]])

    for i in range(img.shape[0]):

        for j in range(img.shape[1]):

            centered\_img[i,j]=img[i,j]\*((-1)\*\*(i+j))

            if centered\_img[i,j]<0:

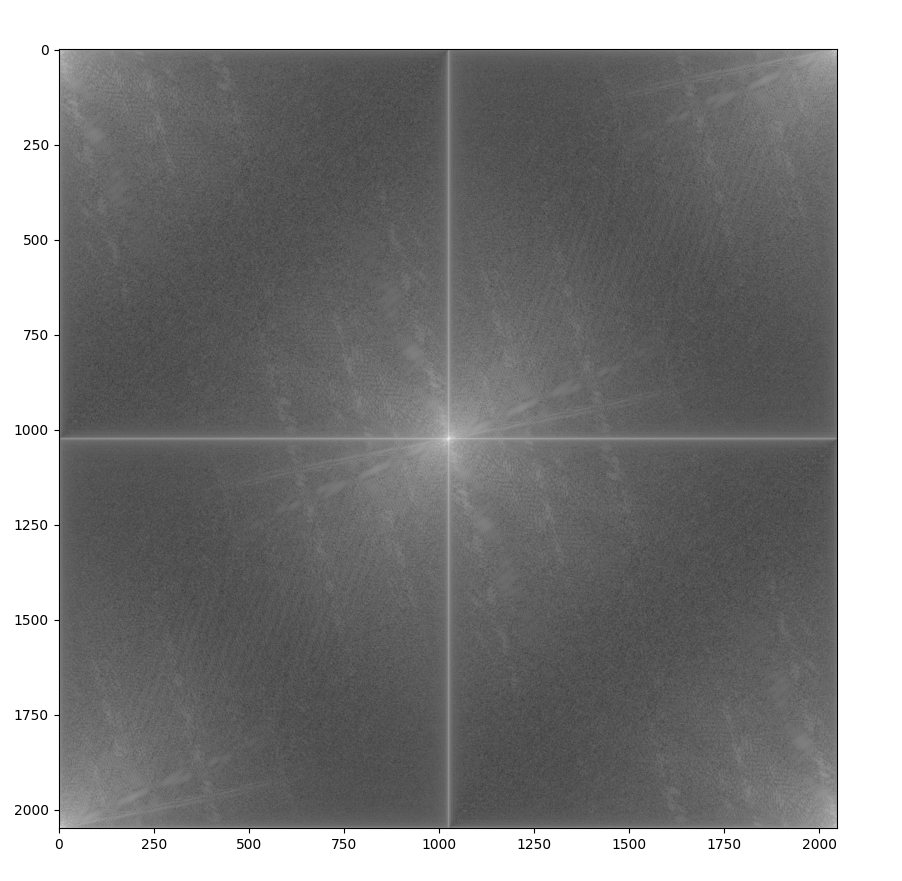
             if centered\_img[i,j]<0:

                centered\_img[i,j]=0

   return centered\_img

دراین تابع ابتدا تمامی مقادیر در (-1)x+y ضرب شدند و سپس مقادیر کوچکتر از صفر به صفر مپ شدند.

خروجی این مراحل درشکل ‏4.1 نشان داده شده است.



شکل ‏4.1 تصاویر خروجی این بخش

## بخش 3)

def ideal\_fil(M,N,cut\_fer):

    H=np.zeros([M,N]).astype('float')

    for u in range(M):

        for v in range(N):

            D=np.sqrt((u-M/2)\*\*2+(v-N/2)\*\*2)

            if D>cut\_fer:

                H[u,v]=0

            else :

                H[u,v]=1

    return H

def butter\_fil(M,N,n,cut\_fer):

    H=np.zeros([M,N]).astype('float')

    for u in range(M):

        for v in range(N):

            D=np.sqrt((u-M/2)\*\*2+(v-N/2)\*\*2)

            H[u,v]=1/(1+pow((D/cut\_fer),2\*n))

    return H

def Gaussian\_fil(M,N,cut\_fer):

    H=np.zeros([M,N]).astype('float')

    for u in range(M):

        for v in range(N):

            D=np.sqrt((u-M/2)\*\*2+(v-N/2)\*\*2)

            H[u,v]=exp((-1\*pow(D,2))/(2\*pow(cut\_fer,2)))

    return H

در این بخش توابع خواسته شده پیاده سازی شدند. در تمامی توابع M,N ابعاد تصویر و cut\_fer شعاع است. در فیلتر باترورث n درجه فیلتر است.

## بخش4)

row,col=shoulder\_img.shape

ideal\_HP\_30=1-ideal\_fil(2\*row,2\*col,30)

ideal\_HP\_100=1-ideal\_fil(2\*row,2\*col,100)

butter\_HP\_30=1-butter\_fil(2\*row,2\*col,2,30)

butter\_HP\_100=1-butter\_fil(2\*row,2\*col,2,100)

gaussian\_HP\_30=1-Gaussian\_fil(2\*row,2\*col,30)

gaussian\_HP\_100=1-Gaussian\_fil(2\*row,2\*col,100)

در این قسمت توابع بالاگذر پیاده سازی شدند. علت استفاده از ضریب 2 در ابعاد، به خاطر پد کردن تصاویر است که ابعاد را بالا برده است.

image\_ideal\_30=ideal\_HP\_30\*shoulder\_img\_dft

image\_ideal\_100=ideal\_HP\_100\*shoulder\_img\_dft

image\_butter\_30=butter\_HP\_30\*shoulder\_img\_dft

image\_butter\_100=butter\_HP\_100\*shoulder\_img\_dft

image\_gauss\_30=gaussian\_HP\_30\*shoulder\_img\_dft

image\_gauss\_100=gaussian\_HP\_100\*shoulder\_img\_dft

دراینجا فیلتر‌ها برروی تصاویر اعمال شدند.

'''ideal'''

shoulder\_IHP\_30=np.fft.ifft2(image\_ideal\_30)

shoulder\_IHP\_30\_show=np.abs(shoulder\_IHP\_30)

shoulder\_IHP\_30\_show=shoulder\_IHP\_30\_show[0:1024,0:1024]

shoulder\_IHP\_100=np.fft.ifft2(image\_ideal\_100)

shoulder\_IHP\_100\_show=np.abs(shoulder\_IHP\_100)

shoulder\_IHP\_100\_show=shoulder\_IHP\_100\_show[0:1024,0:1024]

'''butterworth'''

shoulder\_BHP\_30=np.fft.ifft2(image\_butter\_30)

shoulder\_BHP\_30\_show=np.abs(shoulder\_BHP\_30)

shoulder\_BHP\_30\_show=shoulder\_BHP\_30\_show[0:1024,0:1024]

shoulder\_BHP\_100=np.fft.ifft2(image\_butter\_100)

shoulder\_BHP\_100\_show=np.abs(shoulder\_BHP\_100)

shoulder\_BHP\_100\_show=shoulder\_BHP\_100\_show[0:1024,0:1024]

'''gaussian'''

shoulder\_GHP\_30=np.fft.ifft2(image\_gauss\_30)

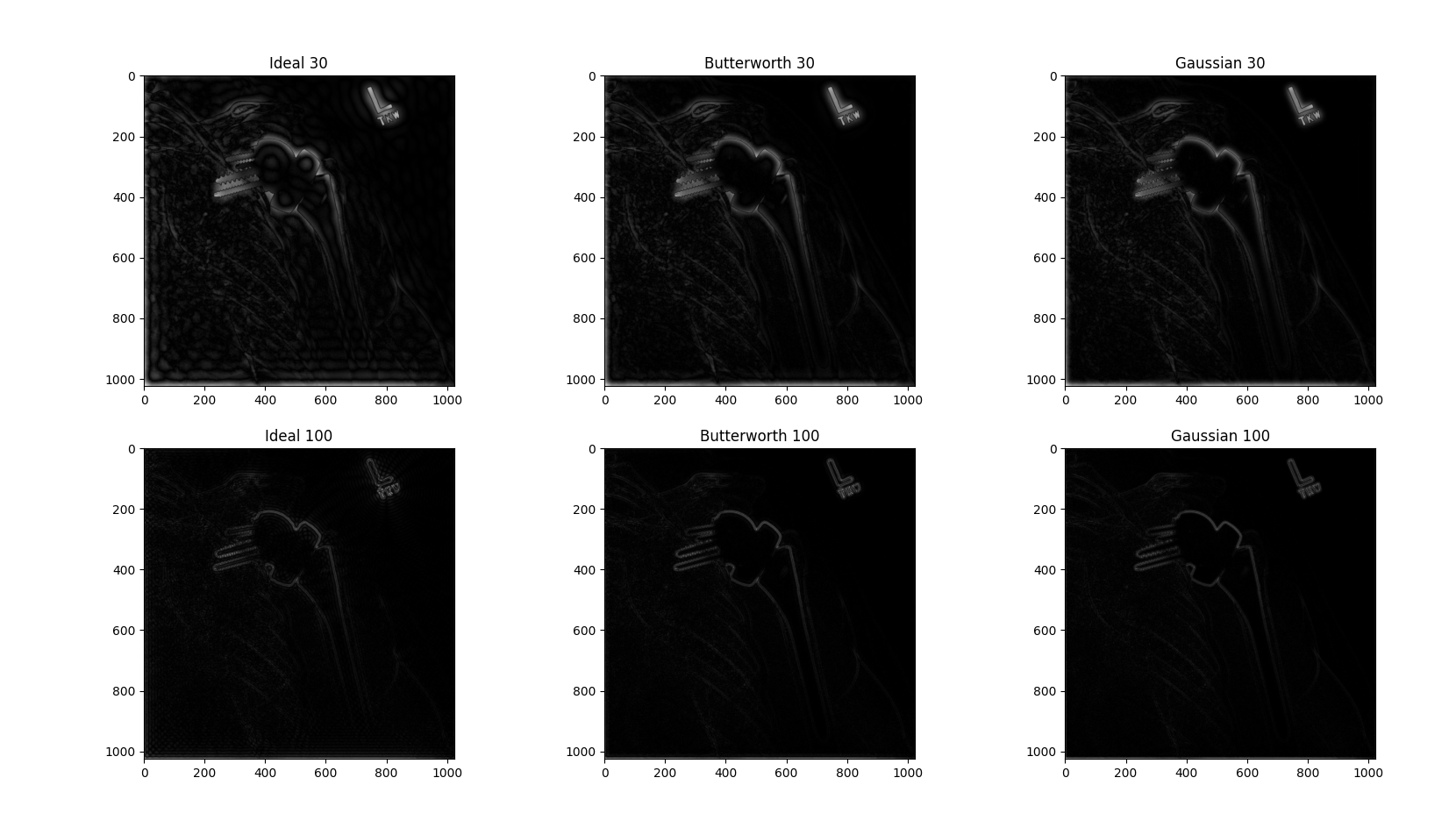
shoulder\_GHP\_30\_show=shoulder\_GHP\_30\_show[0:1024,0:1024]

shoulder\_GHP\_100=np.fft.ifft2(image\_gauss\_100)

shoulder\_GHP\_100\_show=np.abs(shoulder\_GHP\_100)

shoulder\_GHP\_100\_show=shoulder\_GHP\_100\_show[0:1024,0:1024]

در این بخش بازگردانی تصاویر به حوزه مکان و برش و حذف بخش پد شده انجام شده است.خروجی تمام مراحل درشکل ‏4.2 نشان داده شده است.



شکل ‏4.2 خروجی این بخش