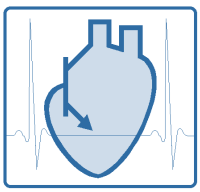


دانشگاه صنعتی امیرکبیر

دانشکده مهندسی پزشکی

گزارش تمرین سوم پردازش تصویر

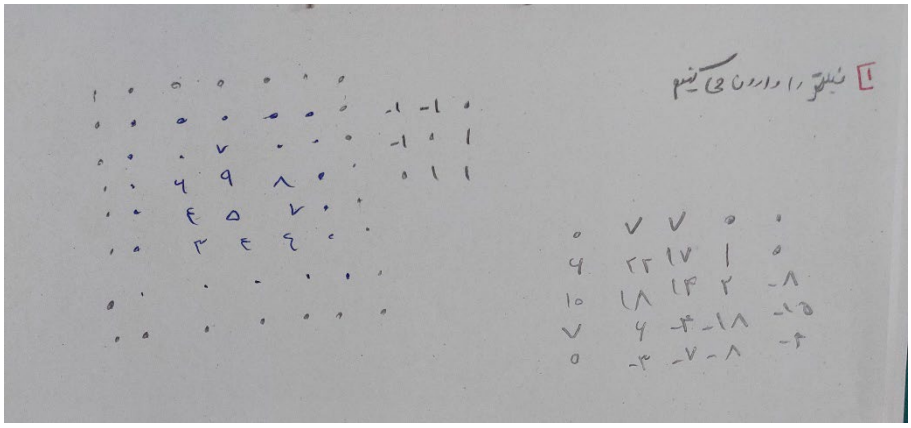
دانشجو

سیدابوالفضل مرتضوی ۹۸۳۳۰۶۳

آذر ۱۴۰۱

فهرست مطالب

سوال اول)



سوال دوم)

۲-۱ بخش الف)

باتوجه به اینکه فیلتر median تمام نویز های ایمپالسی را حذف می‌کند پس ب و ج تصاویری هستند که فیلتر median برروی آنها اعمال شده است. ازآنجا که بزرگتر بودن ابعاد فیلتر باعث مات شدن تصویر می‌شود پس تصویر ج مربوط به فیلتر median باابعاد ۵*۵ است و تصویر ب تصویر این فیلتر با ابعاد ۳*۳ است. باهمین استدلال تصویر آ مربوط به فیلترمیانگین با ابعاد۵*۵ بوده و تصویر د مربوط به این فیلتر با ابعاد ۳*۳ است.

۲-۲ بخش ب)

از آنجا که فیلتر لاپلاسین تصویر را در تمام جهات شارپ کرده و مرز هارا مشخص می‌کند پس تصویر ج تصویر مربوط به این فیلتر است. درمیان فیلتر های رابرت فیلتر x در جهت x تصویر را شارپ می‌کند پس تصویر آ متعلق به این فیلتر است و در آخر تصویر ب برای فیلتر رابرت در جهت y است.

۲-۳ بخش ج)

ازآنجا که مقدار مرکزی فیلتر مثبت است پس فیلتر خودش در ۱- ضرب شده و برای شارپ شدن مرز ها در این مورد باید این حاصل را باتصویر جمع کنیم. پس تصویری که لبه های شارپ تر دارد یعنی تصویر آ تصویری است که با فیلتر جمع شده است.

سوال سوم)

۱-۳ بخش الف)

```
def correlation(f, w, kernel_size):
    out = np.zeros(np.shape(f))
    f = np.pad(f,1)
    for i in range(f.shape[0]-kernel_size+1):
        for j in range(f.shape[1]-kernel_size+1):
            out[i, j] = np.sum(f[i:i+kernel_size, j:j+kernel_size]*w)
    return out

def img_filter(image, filter_name, kernel_size=3):

    if filter_name == 'Mean':

        filter_matrix = np.ones((kernel_size,kernel_size))/kernel_size**2
        out = correlation(image, filter_matrix, kernel_size)

    elif filter_name == 'Median':

        out = np.zeros(np.shape(image),dtype = 'int16')
        image = np.pad(image,1)
        for i in range(image.shape[0]-kernel_size+1):
            for j in range(image.shape[1]-kernel_size+1):
                out[i, j] = np.median(image[i:i+kernel_size,
j:j+kernel_size])

    elif filter_name == 'Sobel_x':

        filter_matrix = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]])
        out = correlation(image, filter_matrix, kernel_size )

    elif filter_name == 'Sobel_y':

        filter_matrix = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]])
```

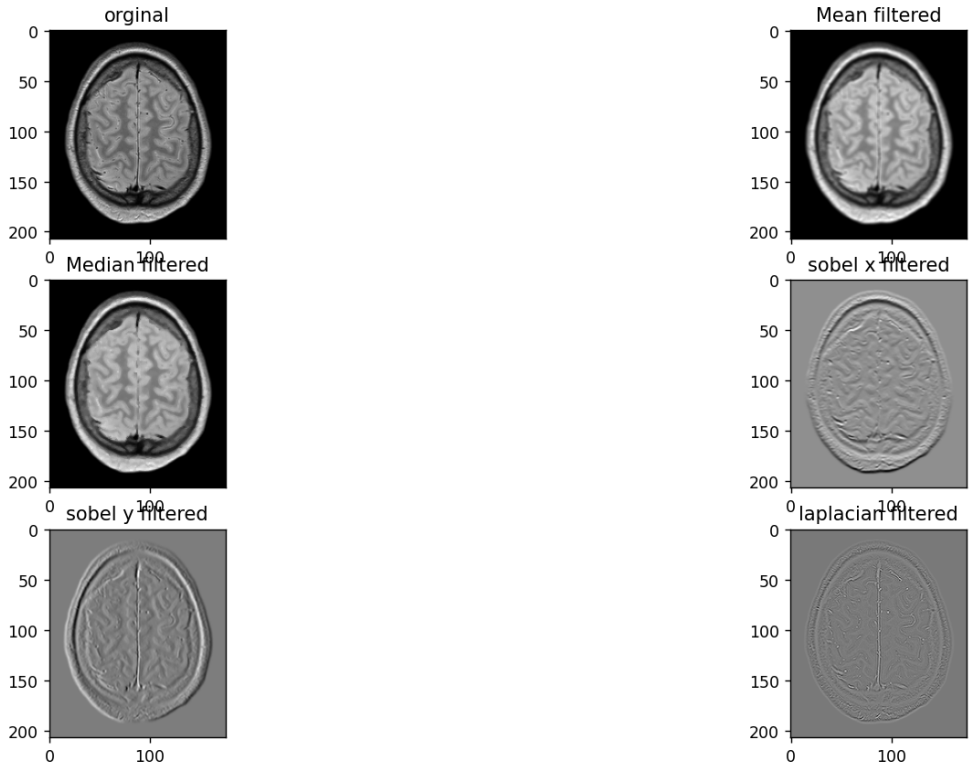
```
        out = correlation(image, filter_matrix, kernel_size )

    elif filter_name == 'Laplacian':

        filter_matrix = np.array([[0, 1, 0], [1, -4, 1], [0, 1, 0]])
        out = correlation(image, filter_matrix, kernel_size
)
    return out
```

در این تمرین ابتدا برای انجام کانولوشون بین کرنل و تصویر تابع correlation تعریف شد. سپس تابع img_filter برای انجام انواع تبدیل‌ها نوشته شد. این تابع ورودی تصویر، نام تبدیل را گرفته و اندازه کرنل سایز پیش‌فرض آن 3*3 است. سپس برای هرتبدیل کرنل برروی تصویر به‌صورت یک واحد یک واحد حرکت کرده و تصویر خروجی به دست‌می‌آید.

۳-۲ بخش ب)

```
MRI_IMG=cv.imread("MRI.png",0)
mean_filtered_img=img_filter(MRI_IMG,'Mean')
median_filtered_img=img_filter(MRI_IMG,'Median')
sobelx_filtered_img=img_filter(MRI_IMG,'Sobel_x')
sobely_filtered_img=img_filter(MRI_IMG,'Sobel_y')
laplacian_filtered_img=img_filter(MRI_IMG,'Laplacian')
plt.subplot(3,2,1)
plt.imshow(MRI_IMG,'gray')
plt.title('orginal')
plt.subplot(3,2,2)
plt.imshow(mean_filtered_img,'gray')
plt.title('Mean filtered')
plt.subplot(3,2,3)
plt.imshow(median_filtered_img,'gray')
plt.title('Median filtered')
plt.subplot(3,2,4)
plt.imshow(sobelx_filtered_img,'gray')
plt.title('sobel x filtered')
plt.subplot(3,2,5)
plt.imshow(sobely_filtered_img,'gray')
plt.title('sobel y filtered')
plt.subplot(3,2,6)
plt.imshow(laplacian_filtered_img,'gray')
plt.title('laplacian filtered')
plt.show()
```

در این بخش تصویر MRI خوانده شده و تبدیل‌های مختلف برروی تصویر اعمال شدند خروجی درشکل ۳.۱ قابل ملاحظه است.



شکل ۳.۱ خروجی مربوط به اعمال فیلترها

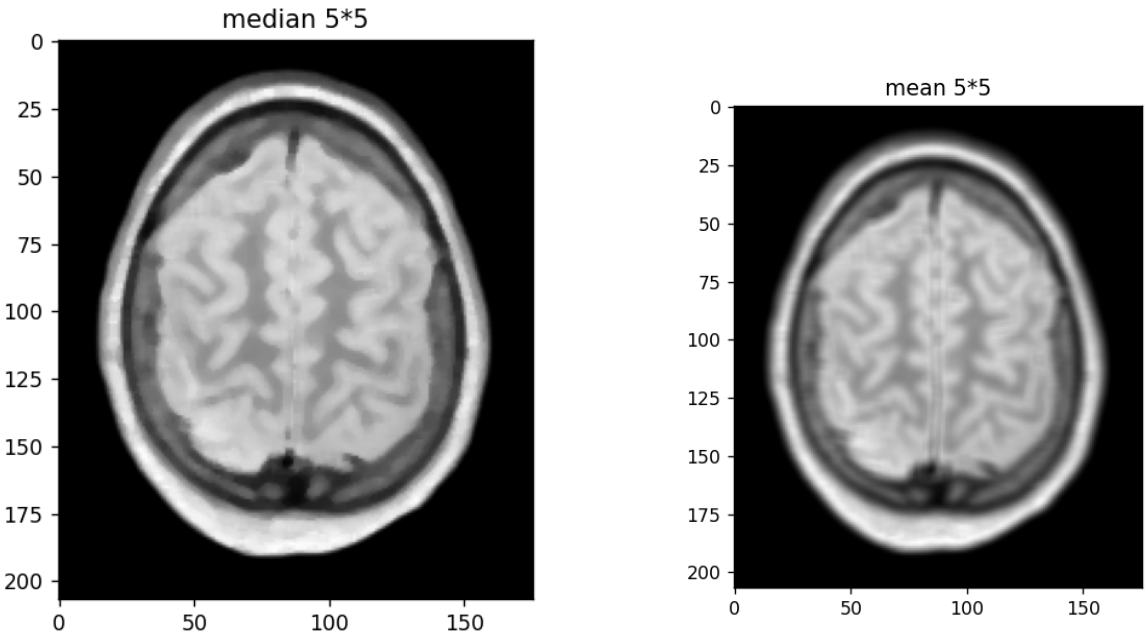
همانطور که قابل مشاهده است تبدیل میانه تصویر را به نوعی شارپ‌تر کرده و برخی بخش‌های پالس مانند را حذف کرده است اما در تصویر میانگین‌گیری شده تصویر نرم تر شده و لبه‌های نرم تری داریم. همچنین در این تبدیل تصویر مات تر شده است.

در تصویر فوق هم قابل مشاهده است که در اعمال فیلتر سوبل در جهت ایکس تصویر تغییر کرده و لبه های تصویر درجهت x قابل ملاحظه تر و شارپتر شده است. در فیلتر سوبل در جهت y لبه‌های تصویر در جهت y شارپتر شده است. همچنین در اعمال این نوع فیلترها به علت خاصیت مشتق ممکن است نویز در تصاویر گسترش پیدا کند.

۳-۳ بخش ج)

در این بخش از تابع ۳-۱ استفاده شد فقط اندازه کرنل پیش فرض از 3*3 به5*5 تغییر یافت.

خروجی این بخش در قابل ملاحظه است.



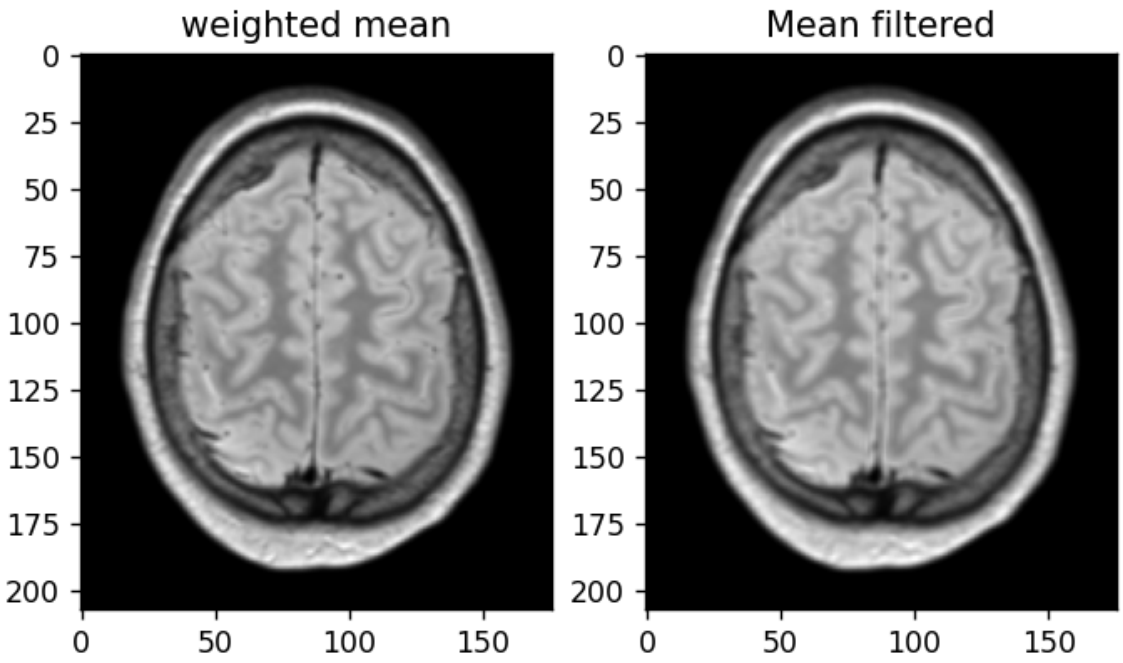
شکل ۳.۲ خروجی مربوط به اعمال فیلترها

در این بخش با اعمال فیلتر با کرنل سایز بزرگتر تصاویر نسبت به تصاویر بخش قبل مات تر شده اند و این اتفاق به خاطر دخیل بودن پیکسل‌های دورتر در تعیین شدت هر پیکسل است.

۳-۴ بخش د)

```
#d
def weighted_mean_filter(image):

    filter_matrix = np.array([[1,2,1],[2,5,2],[1,2,1]])/16
    out = correlation(image, filter_matrix, 3)
    return out
weighted_mean_filtered=weighted_mean_filter(MRI_IMG)
plt.subplot(1,2,1)
plt.imshow(weighted_mean_filtered,'gray')
plt.title('weighted mean')
plt.subplot(1,2,2)
plt.imshow(mean_filtered_img,'gray')
plt.title('Mean filtered')
plt.show()
```

در این بخش تابع weighted_mean_filter تعریف شد. به طوری که همسایگی های نزدیک تر وزن بالاتری نسبت به همسایه های دورتر دارند.خروجی این بخش درشکل ۳.۳ قابل مشاهده است.



شکل ۳.۳ خروجی فیلتر وزن دار و بدون وزن

همانطور که در تصویر فوق قابل مشاهده است با استفاده ازاین نوع فیلتر تصویر کمتر مات شده و این به خاطر تاثیر کمتر پیکسل‌های دور در شدت هرپیکسل است.

سوال چهارم)

۴-۱ بخش آ)

```
noisy_img=cv.imread('Noisy_Spine.png',0)
#a
median_filtered=img_filter(noisy_img,'Median',5)
```

در انجام این بخش از توابع تعریف شده در تمرین قبل استفاده شد.

فیلتر میانه همواره کمک می‌کند تا نویزهای ایمپالسی حذف شوند همچنین ممکن است باعث از بین رفتن آبجکت‌هایی با حالت ایمپالسی شوند. در این مورد با توجه به زیاد بودنن نویزاین فیلتر باعث حذف نویز ها می‌شود.

۴-۲ بخش ب)

```
mean_median_filtered=img_filter(median_filtered2,'Mean')
```

در این بخش هم از توابع تعریف شده در بخش‌های پیشین استفاده شد.

در این بخش با اعمال فیلتر میانگین به تصویر باعث نرم تر شدن گوشه های تصویرمی‌شویم. همچنین اگر نویز هایی برروی تصویر باشند که خاصیت ایمپالسی نداشته باشند بااعمال این فیلتر نرم شده و تا حد امکان اثرشان تضعیف می‌شود.

۴-۳ بخش ج)

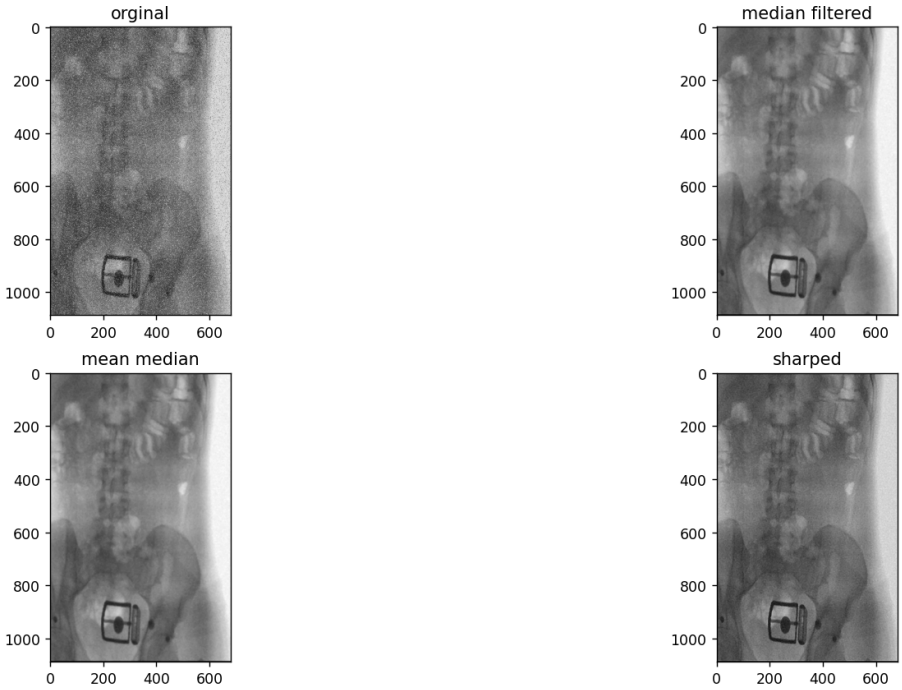
```
laplacian_filtered_image=img_filter(mean_median_filtered2,'Laplacian')
sharped_image=mean_median_filtered-laplacian_filtered_image
(row,col)=sharped_image.shape
for i in range(row):
    for j in range(col):
        if sharped_image[i,j]>255:
            sharped_image[i,j]=255
        elif sharped_image[i,j]<0:
            sharped_image[i,j]=0
```

بااستفاده از توابع قبلی فیلتر لاپلاسین برروی تصویر اعمال شد. و سپس مقادیر به بازه ۰و۲۵۵ مپ شدند.

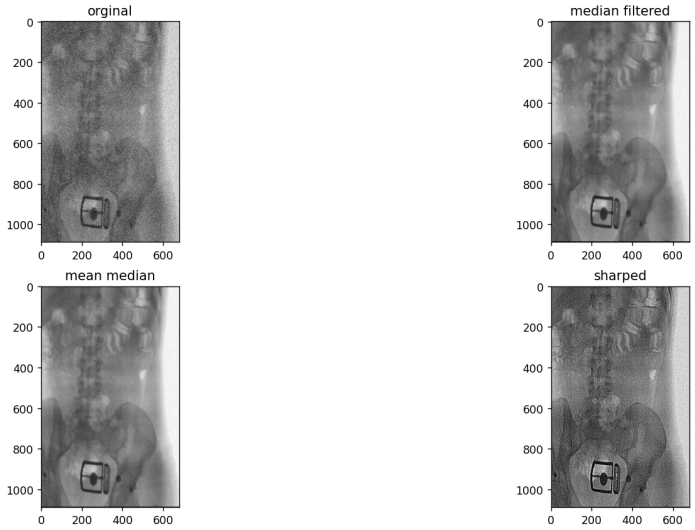
در این بخش باید از علامت منفی برای اعمال فیلتر لاپلاس برروی تصویر استفاده کنیم. زیرا علامت عنصر وسط منفی است. باانجام این کار لبه‌های تصویر که در مرحله قبلی نرم شده بودند در این مرحله دوباره شارپترمی‌شوند. برای جمع کردن ضریب c را برابر ۱- قراردادم زیر با تغییر ضریب و بزرگتر کردن آن نویز

های باقی مانده در تصویر به شدت تقویت می‌شدند و با کوچکتر کردن ضریب لبه ها به اندازه کافی شارپ نمی‌شدند.

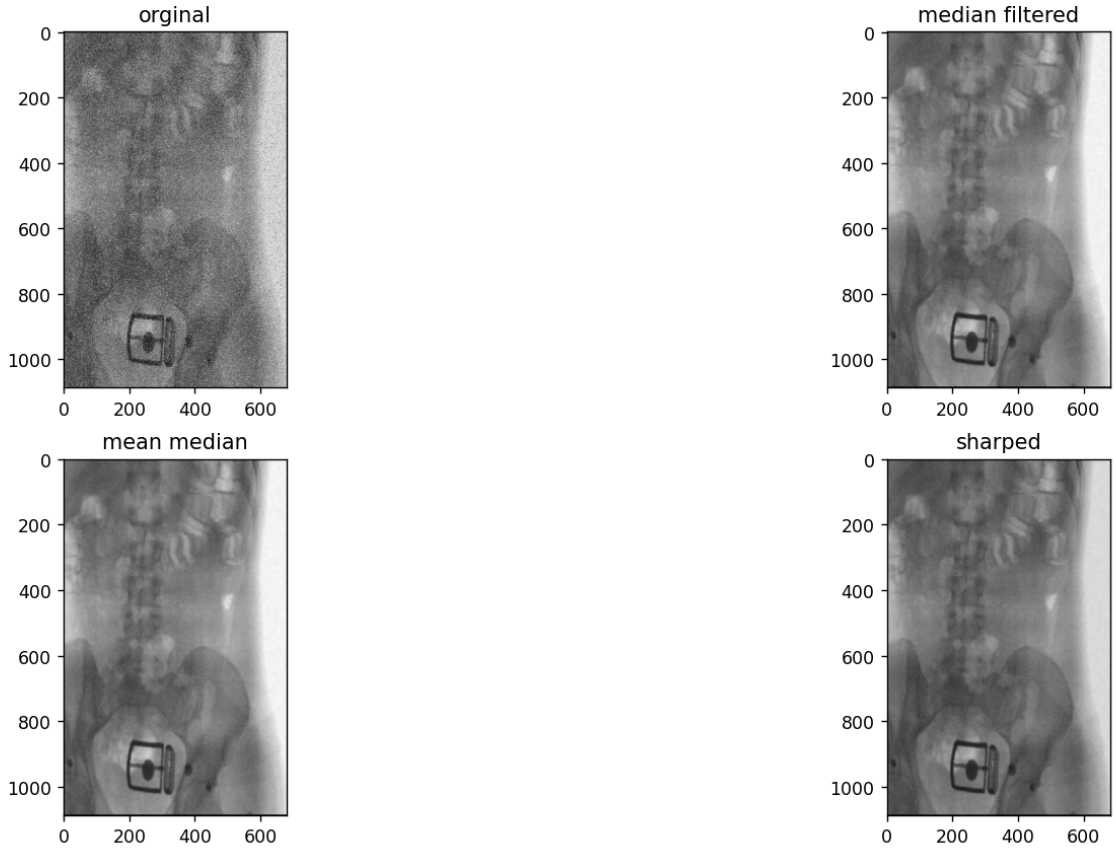
۴-۴ بخش د)



شکل ۴.۱ خروجی بخش های آ تا ج با ضریب ۱-



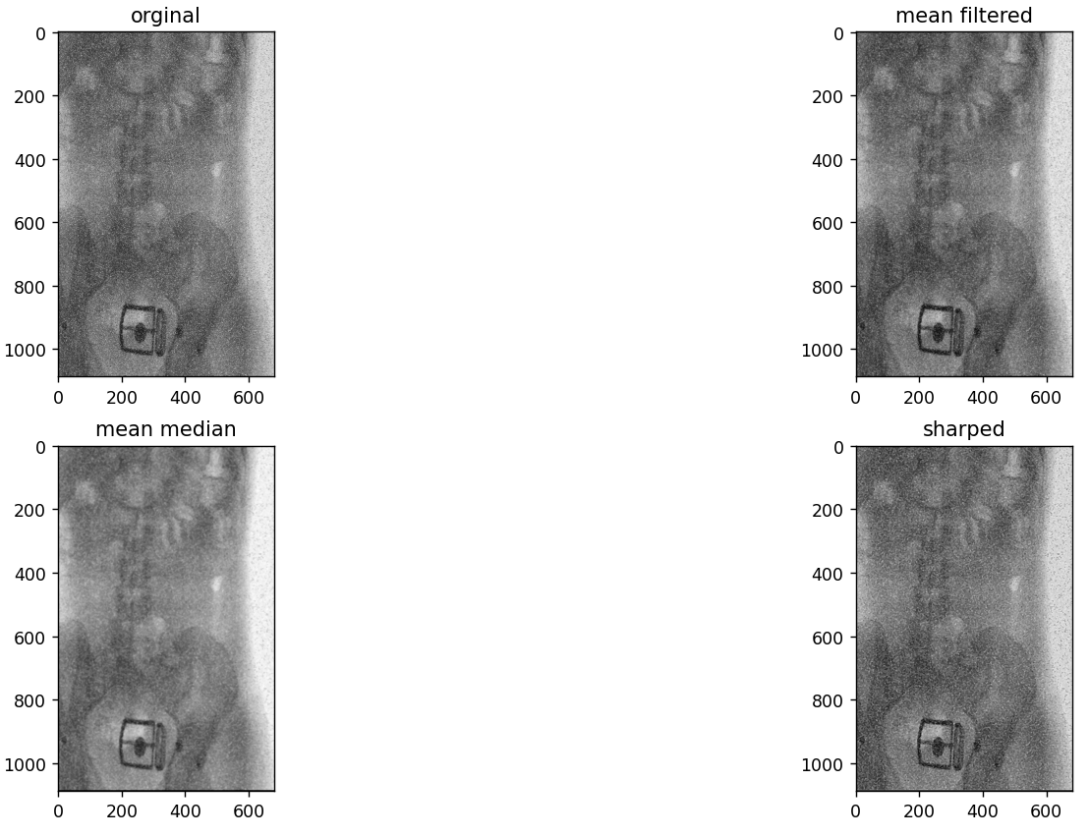
شکل ۴.۲ ضریب ۵-



شکل ۴.۳ضریب ۰.۵-

۴-۵ بخش ه)

بله خروجی این دو بخش تفاوت هایی باهم دارد چراکه اعمال فیلتر میانگین‌گیری درابتدا باعث نرم شدن نویز ها می‌شود و فیلتر میانه نمی‌تواند تمام نویز هارا به خوبی حذف کند. پس نویز در تصویر باقی می‌مانند. خروجی این بخش در شکل ۴.۴قابل مشاهده است.



شکل ۴.۴ خروجی بخش ه با ضریب ۱-