باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس سیگنال ها و سیستم ها

گزارش تمرین سری سوم متلب

علی محرابیان

96102331

استاد

دکتر کربلایی آقاجان

1.یافتن لبه درتصاویر

```
function y=sobel(m)

    Gx=[-1 0 1; -2 0 2; -1 0 1];
    Gy=[-1 -2 -1; 0 0 0; 1 2 1];


    mf=fft2(m);
    r1=size(mf,1);
    r2=size(mf,2);

    gx=fft2(Gx,r1,r2);
    gy=fft2(Gy,r1,r2);

     s11=mf.*gx;
     s22=mf.*gy;

     s1=ifft2(s11);
     s2=ifft2(s22);

        k=sqrt(s1.^2+s2.^2);
```

درابتدا روش کلی کار را توضیخ می دهیم.

ابتدا به کمک دستور fft2،تصویر وکرنل را

به حوزه فرکانس برده وبه کمک قضیه کانولوشن،

دو ماتریس را نقطه به نقطه درهم ضرب کرده و

پس از انجام اعمال مطلوب درهر روش،

تصویر نهایی را به حوزه قبل برمی گردانیم.

به کمک دستور tic,toc زمان پیاده سازی دوالگوریتم رامقایسه می کنیم.

در الگوریتم sobel،زمان پیاده سازی به صورت روبرو است.

```
Elapsed time is 0.633928 seconds.
```

درالگوریتم kirsch،زمان پیاده سازی به صورت روبرو می باشد.

```
Elapsed time is 2.486122 seconds.
```

ابتدا خروجی یکی ازتصاویر را با هردو الگوریتم درکنارهم مشاهده می کنیم.

الگوریتم sobel

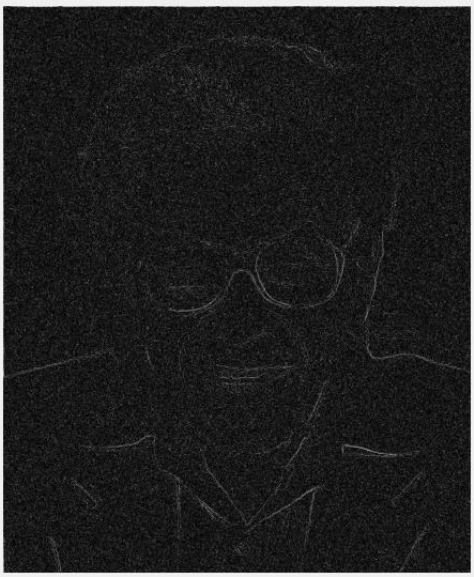
الگوریتم kirsch

KIRSCH

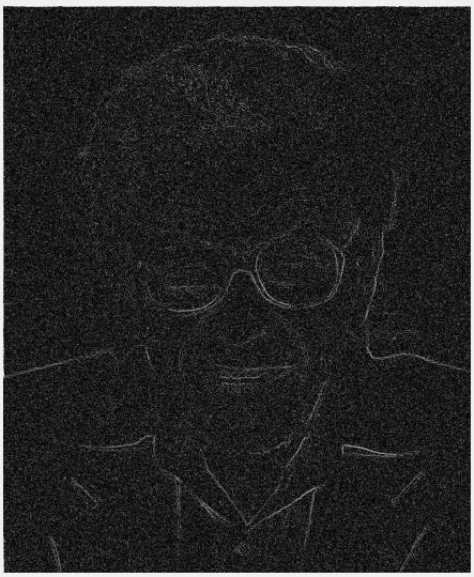
SOBEL

درحالت معمولی خیلی تفاوتی مشاهده نمی شود ولی الگوریتم sobel درلبه های تیزتر،عملکرد کمی بهتر دارد.

حال تصویراصلی را به نویز آغشته می کنیم.یک نویزگاوسی با میانگین 0 و واریانس 0.048 به تصویر اضافه می کنیم.خروجی ها به صورت زیر هستند.



KIRSCH

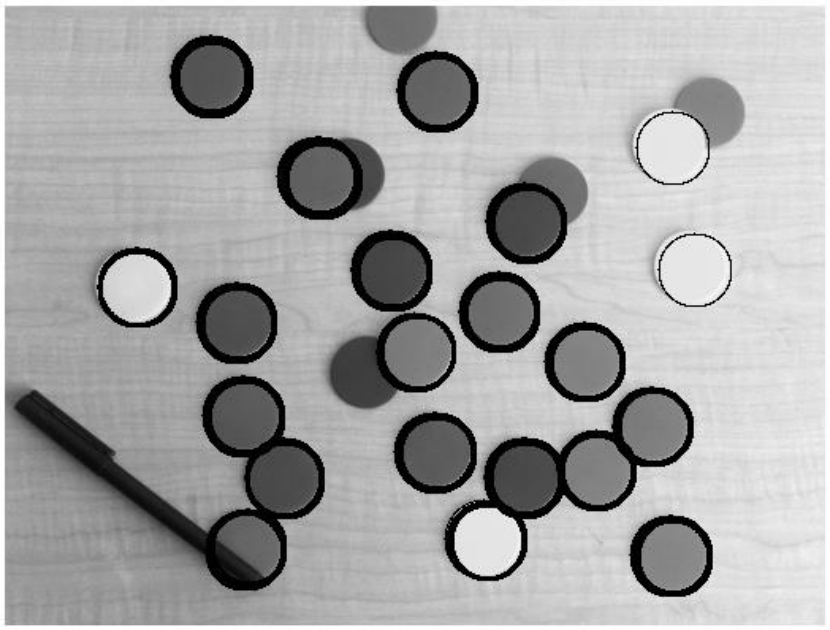


SOBEL

مشاهده می شود که خروجی الگوریتم sobel دراین حالت بهتر و وضوح بیشتری دارد.پس به طورکلی بسته به کار وخروجی که مدنظرماست،می توان از هردوالگوریتم استفاده کرد.

2.الگوریتم برای شمارش دایره ها در تصاویر

دراین روش به طور کلی ابتدا به کمک تابع circle وتعریف maxx و min ،دایره هایی با شعاع های در این بازه می سازیم .سپس ماتریس با تابع edgeE،که خودمان تعریف کردیم،لبه ها را پیدا می کنیم. سپس با کانولوشن این ماتریس در دایره ها،لبه های دایره ها را پیدا می کنیم.در تصاویر زیر می توان تصاویر را مشاهده کرد.



3.اهمیت فاز و دامنه

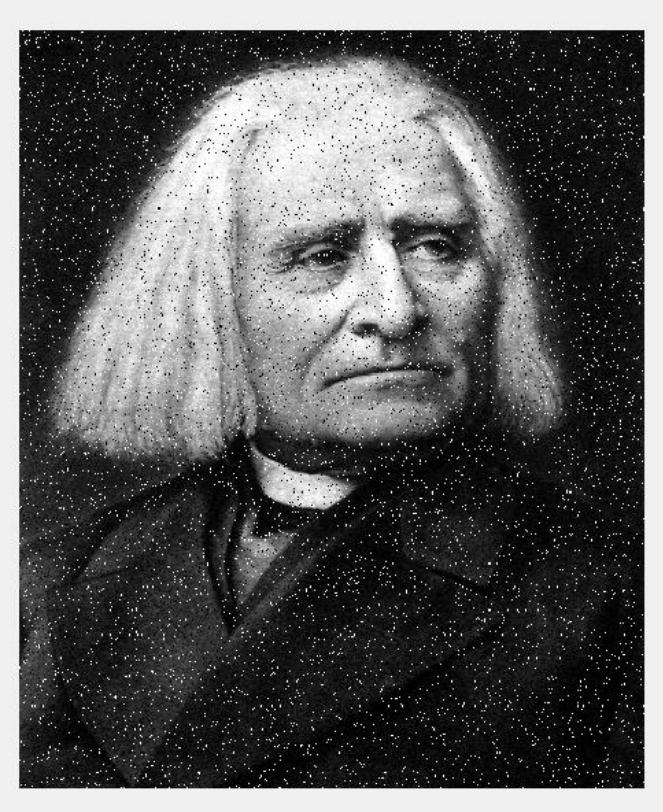
عملیات خواسته شده را انجام می دهیم.خروجی به صورت زیر می باشد.

همان طور که مشاهده می شود،فاز تصویر اطلاعات مهم تری را در بر دارد.

به طورکلی تصاویر دارای feature و ویژگی های مشخصی مانند لبه ها،گوشه ها و... هستند.فاز تصویر اطلاعات درباره مکان این ویژگی ها را دربر دارد درحالی که دامنه،اندازه ومشخصات این Feature ها را دارد.

4.حذف نویز

1.نویز salt and pepper یا نویز impulse،به صورت نقاط سیاه و سفید برروی پیکسل های عکس ظاهر می شود.پیکسل های روشن به تاریک و بالعکس تبدیل می شوند.از روش های های مرسوم برای حذف این نویز،استفاده از فیلترمیانه گیر است که درادامه شرح داده خواهد شد.

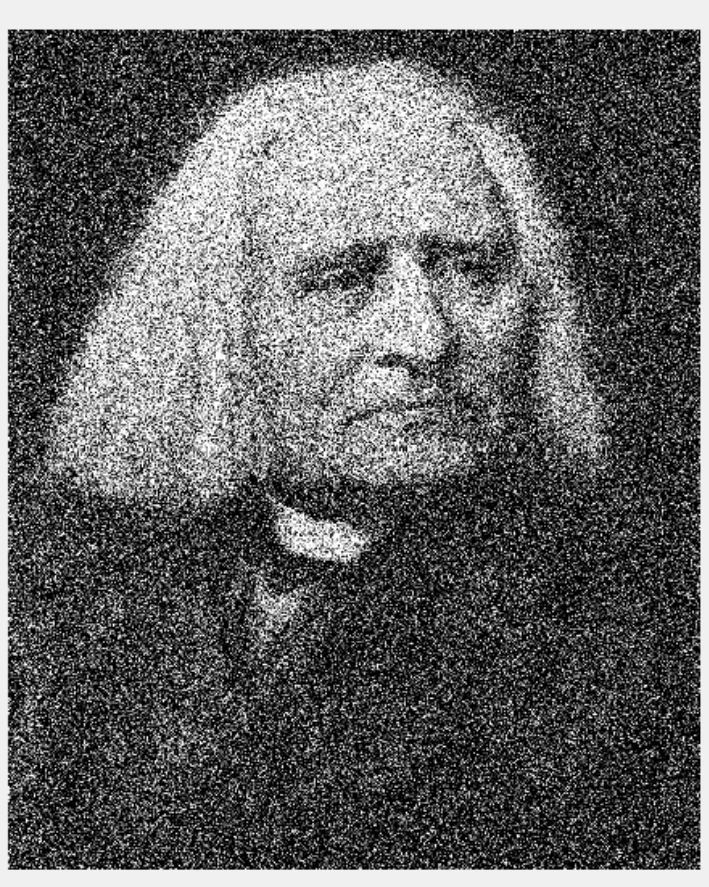


2.نویز گاوسی که یک نویز استاتیکی است و از توزیع احتمالاتی زیر پیروی می کند.

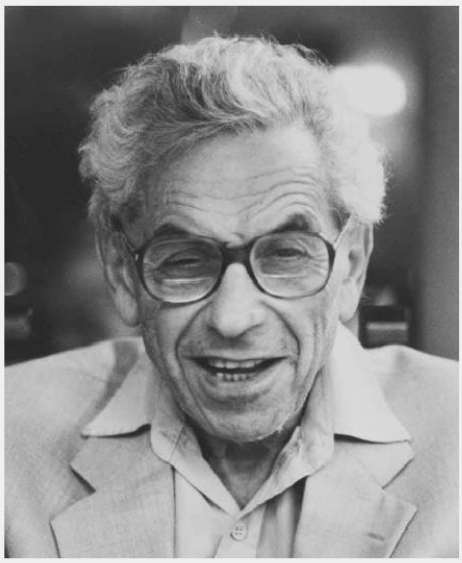
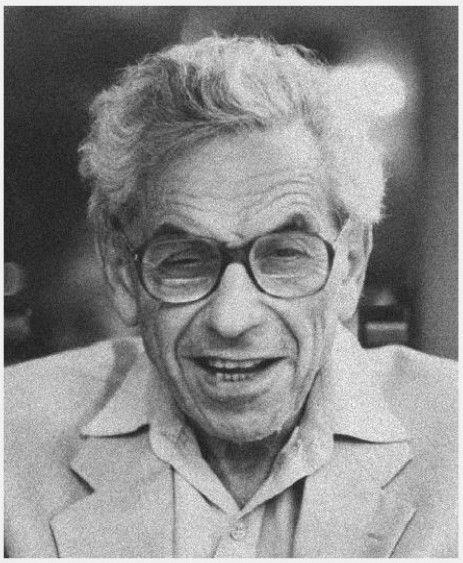
$$p(g) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(g-\mu)^2}{2\sigma^2}}$$

که σ بیانگر انحراف معیار و μ نشان دهنده میانگین است.

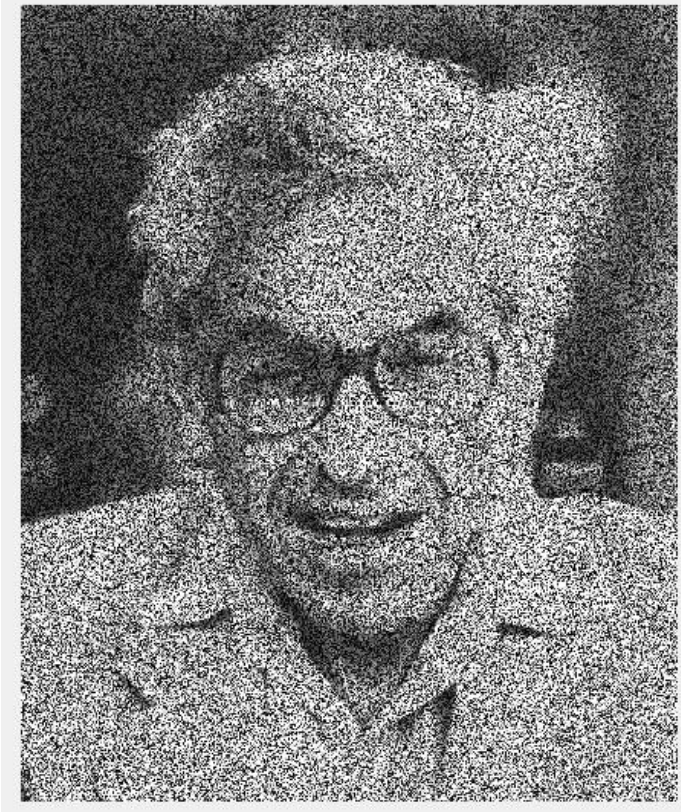
این نویز باعث می شود که تصاویر کم رنگ وتار به نظر برسند ومعمولا مقادیر خاکستری در تصاویر دیجیتال را مختل می کند.در این مورد از فیلتر گاوسی که در ادامه تشریح می شود،برای حذف این نویز استفاده می کنیم.



3.نویز پواسون یا نویز shot،معمولا دردستگاه های الکتریکی هنگامی اتفاق می افتد که در شمارش فوتون های یک پیکسل،اختلال رخ دهد.

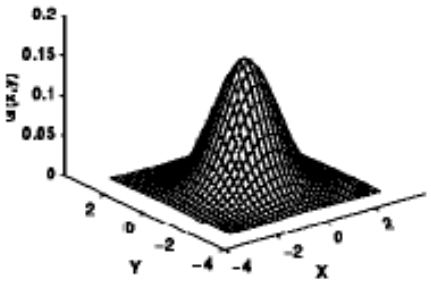
4.نویز اسپکل یک آشفته ای نقطه ای است که معمولا به عنوان نویز ضرب شونده درتصاویر مدل می شود.این نویز درتصاویر رادار است و وابسته به سیگنال بوده وبه دلیل نوسانات فازسیگنال های بازگشتی امواج الکترومغناطیسی به وجود می آید.



ابتدا نحوه عملکرد فیلتر گاوسی را توضیح می دهیم.

چون تصاویر ما دوبعدی هستند،در نتیجه باید از تابع گاوسی دوبعدی استفاده کنیم.

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{\frac{x^2+y^2}{2\sigma^2}}$$



فیلتر گاوسی به صورت فیلترپایین گذر عمل می کند.ما باید یک کرنل که توزیع گاوسی دارد،تولید کنیم که ضرایب هرچه از مرکز دورمی شویم،کاهش می یابند.میانگین صفر وتوزیع متقارن است.معمولا این فیلتر روشنایی را حفظ نمی کند.

به طور مثال برای کرنل سایز 5*5،ورودی های x ,y را به صورت زیرتعریف می کنیم.

```
x =

    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
```

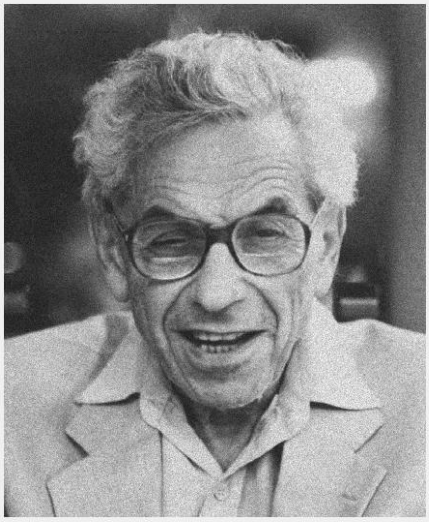
```
y =

    -2    -2    -2    -2    -2
    -1    -1    -1    -1    -1
     0     0     0     0     0
     1     1     1     1     1
     2     2     2     2     2
```

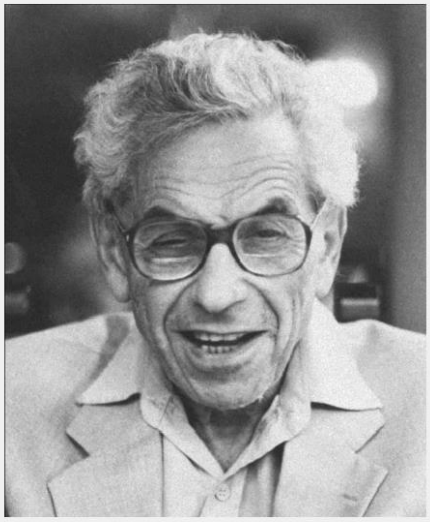
سپس همراه با واریانس تعریف شده،کرنل به دست آمده و ازکانولوشن آن با عکس،خروجی فیلتر شده به دست می آید.طبق راهنمایی،0 هایی به گوشه های تصویر اضافه شد تا از خروج عکس از کادر جلوگیری کنیم.به ترتیب اثر این فیلتر را روی عکس های آلوده به نویز مشاهده می کنیم.

سایز کرنل برابر با 5 و واریانس برابربا 2 انتخاب شد.

سایز کرنل برابر با 5 و واریانس برابربا 2 انتخاب شد.دراین مورد خروجی خوبی را شاهد هستیم.

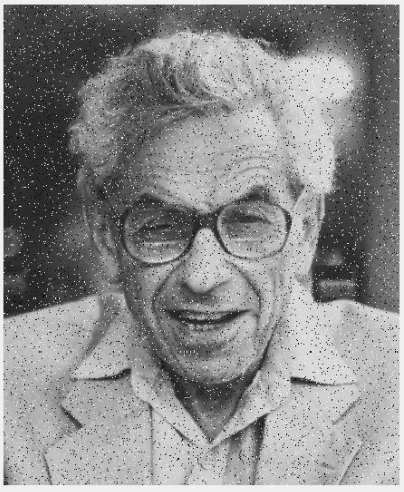


POISSON NOISE

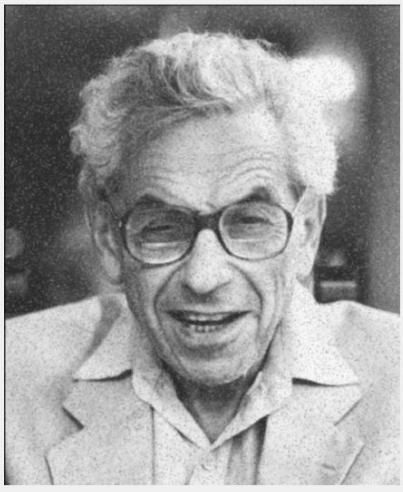


FILTERED

سایز کرنل برابر با 3 و واریانس برابربا 2 انتخاب شد.دراین مورد خروجی خوبی را شاهد هستیم.



SALT&PEPPER NOISE



FILTERED

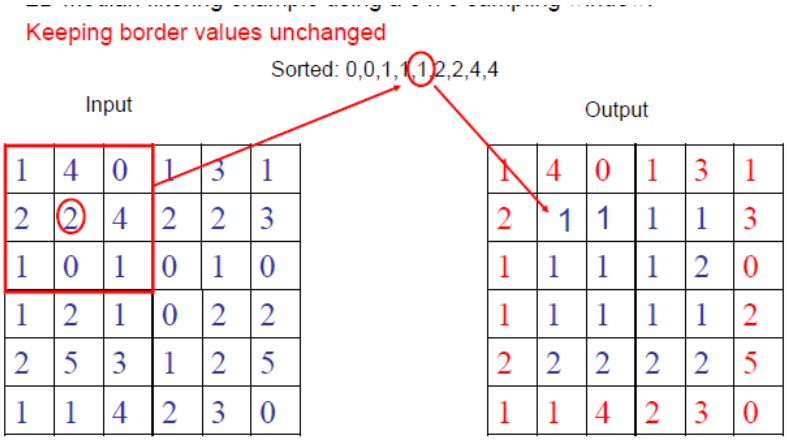
سایز کرنل برابر با 5 و واریانس برابربا 2.5 انتخاب شد.

می توان دید که فیلتر گاوسی در مواجه با نویزهای Gaussain,poisson عملکرد خوبی دارد.

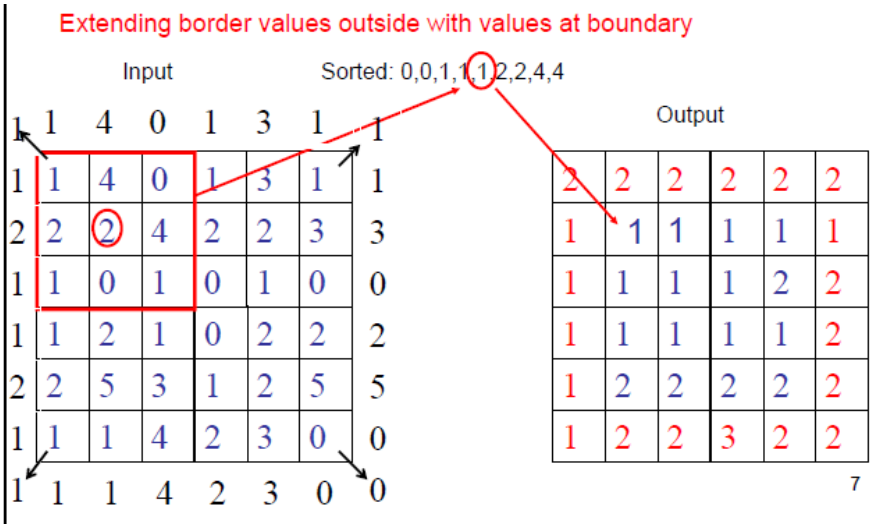
حال فیلتر میانه گیر را توضیح می دهیم.

یک فیلتر غیرخطی است که هر پیکسل را با میانه خانه های اطرافش جایگزین می کند.در این روش درگوشه های تصویر به مشکل می خوریم که به 3طریق زیر قابل حل می باشد.

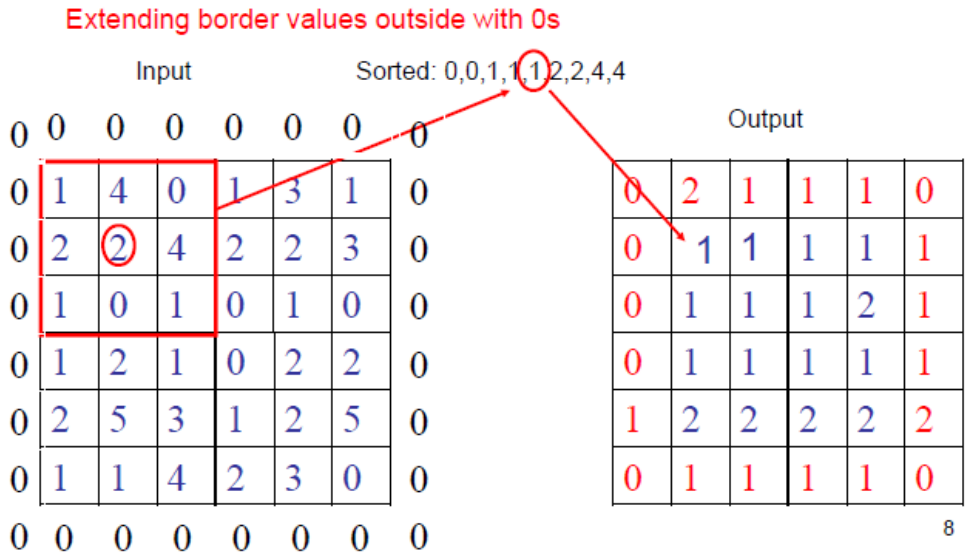
در روش اول،پیکسل های کناری را نگه می داریم و میانه گیری را روی بقیه پیکسل ها انجام می دهیم.



درروش دوم،تصویر را با پیکسل های کناری Extend می کنیم وسپس میانه گیری را انجام می دهیم.

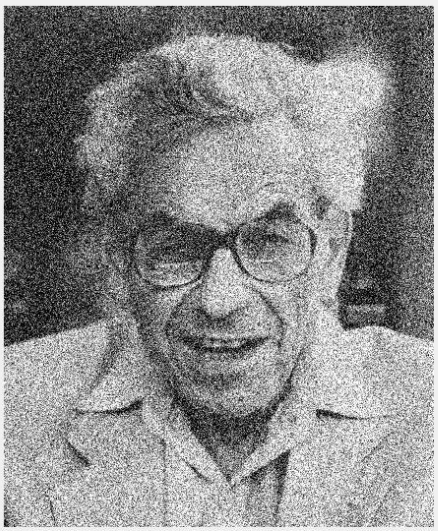


درروش سوم،خانه های کناری را با Extend،0 می کنیم وسپس میانه گیری را انجام می دهیم.

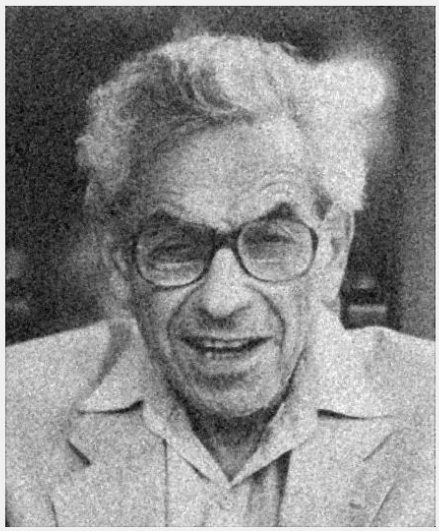


ما روش سوم را پیاده سازی کردیم.به ترتیب اثرفیلتر را روی عکس های آلوده به نویزمشاهده می کنیم.

سایزکرنل را برابر با 5 انتخاب کردیم.

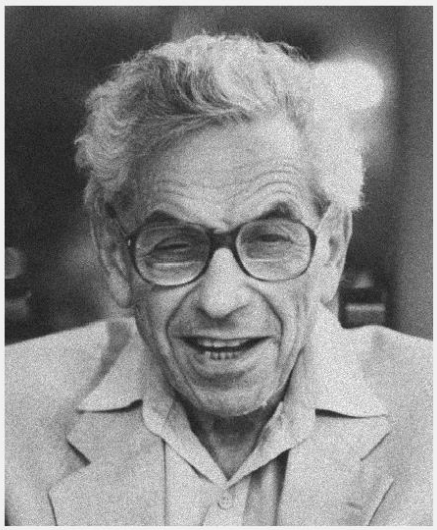


GAUSSIAN NOISE

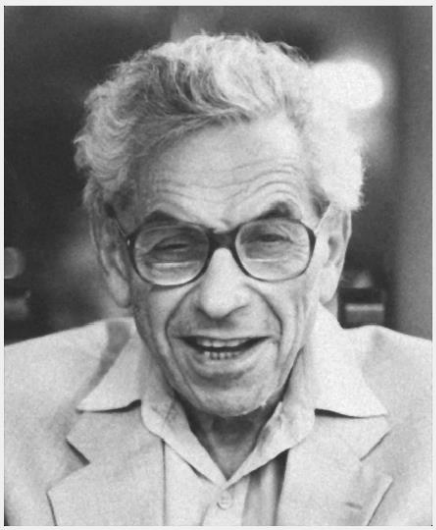


FILTERED

سایزکرنل را برابر با 3 انتخاب کردیم.

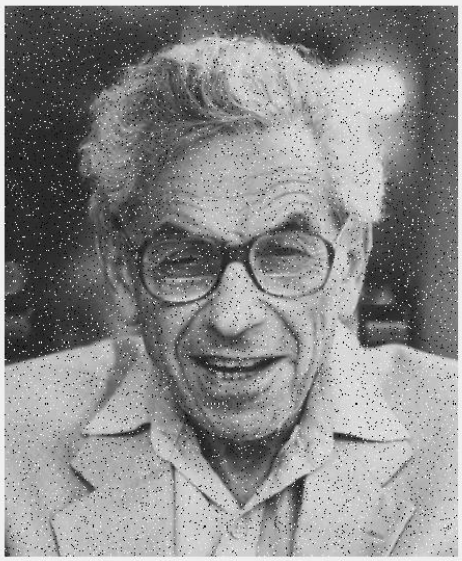


POISSON NOISE

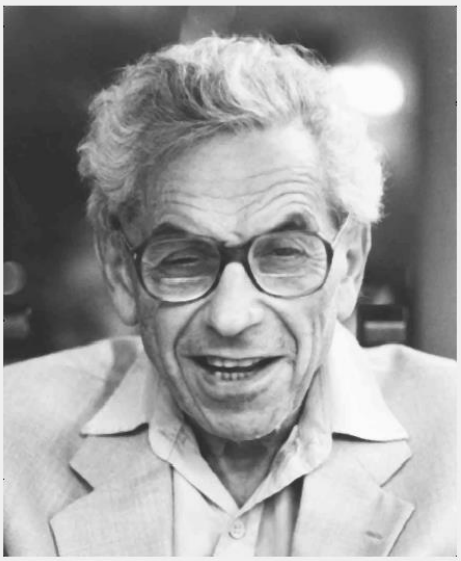


FILTERED

سایزکرنل رابرابر با 3انتخاب کردیم.خروجی خوبی را مشاهده می کنیم.



SALT&PEPPER NOISE



FILTERED

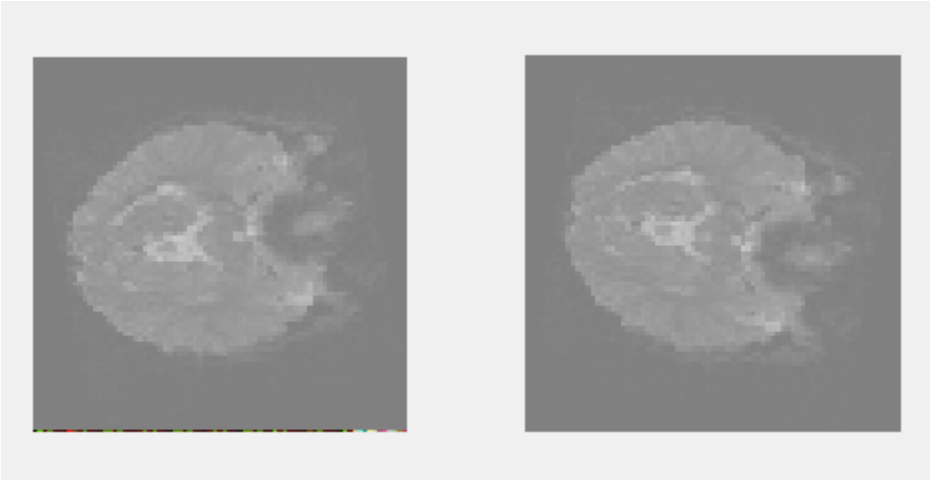
سایز کرنل را برابر 5 انتخاب کردیم.خروجی فوق العاده ای را شاهد هستیم.

می توان فهمید که برای نویزpoisson و به خصوص salt & pepper،این فیلتر عملکرد خوبی دارد.

5.تصاویر FMRI

دراین قسمت هدف ما پیدا کردن بهترین پارامتر های دوران وانتقال برای align کردن دو تصویر است.

ابتدا این کاررا به کمک optimizer متلب وسپس خودمان انجام می دهیم.در ابتدا دو تصویر را نمایش می دهیم.



تصویر چپ راثابت نگه داشته وتصویر سمت راست را تغییر می دهیم.

```
[optimizer, metric] = imregconfig('multimodal');
optimizer.InitialRadius = 0.004;
optimizer.Epsilon = 1.5e-7;
optimizer.GrowthFactor = 1.001;
optimizer.MaximumIterations = 800;
tform = imregtform(moving, fixed, 'affine', optimizer, metric);
movingRegistered = imwarp(moving,tform,'OutputView',imref2d(size(fixed)));
```

پارامتر ها به صورت روبرو هستند.

```
angle =

   19.8184

>> x(1)

ans =

  -10.3416
```

```
>> y(1)

ans =

   15.2086

>> scale

scale =

    1.0030
```

$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = (X\ Y\ 1) \quad * \begin{pmatrix} s\cos\theta & -s\sin\theta & 0 \\ s\sin\theta & s\cos\theta & 0 \\ a & b & 1 \end{pmatrix}$$

حال ما s=1 فرض کرده و پارامترهای بالا را تغییر می دهیم تا ماکزیمم correlation بین دو تصویر را پیدا کینم.

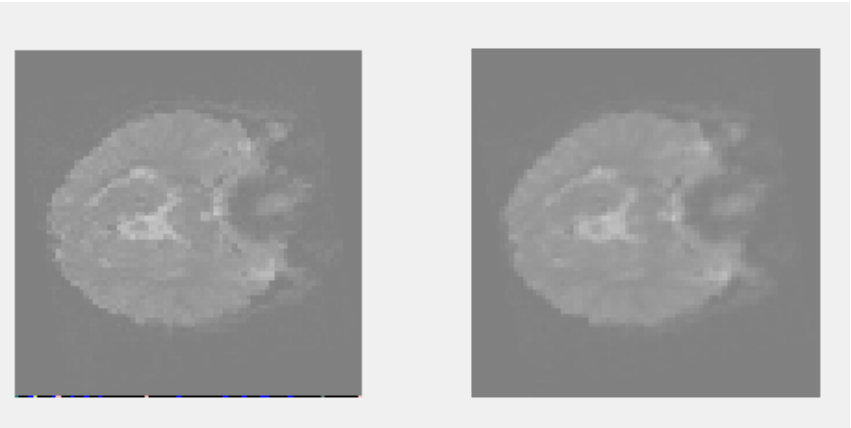
$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \overline{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \overline{B})^2\right)}}$$

where $\overline{A}$ = mean2(A), and $\overline{B}$ = mean2(B).

پارامترها به صورت روبرو هستند.

3957	-11	15.5000	20	0.9753

خانه اول،انتقال در راستای x،خانه دوم انتقال در راستای y،خانه سوم زاویه دوران وخانه آخر،ضریب همبستگی بین دو تصویر است.



می بینیم که دو تصویر تقریبا مشابه هستند.