الضجيج:

يمكن أن نصنع الضجيج عند دراسة الصور ومعالجتها في ماتلاب من خلال التعليمة

J = imnoise(I , type)

I: The name of the image

Type : ‘gaussian’ Gaussian white noise

‘poisson’: Poisson noise

‘salt & pepper’: On and off pixels

الفلترة:

الفلترة هي تنقية لتعديل وتحسين الصورة, وهي عملية من عمليات الـ neighborhood operation حيث نقوم هنا بتطبيق خوارزمية معينة لأجل حساب قيمة كل بكسل ناتج في صورة الخرج من البكسل المقابل له والبكسلات الأخرى المجاورة في صورة الدخل وهكذا يختلف نوع عملية الفلترة وفقا لنوع الخوارزمية المطبقة.

أنواع الفلترة:

Mean Filter -1

```
h = fspecial('average', hsize)
h = fspecial('disk', radius)
h = fspecial('gaussian', hsize, sigma)
h = fspecial('laplacian', alpha)
h = fspecial('log', hsize, sigma)
h = fspecial('motion', len, theta)
```

نقوم ببناء هذه الفلاتر من خلال التعليمة

```
h = fspecial( type ,  parameters )
```

ثم نقوم بعملية الفلترة من خلال التعليمة:

```
B = imfilter(A,H)
B = imfilter(A, H, option1, option2,...)
```

A: هي المصفوفة المعبرة عن الصورة المراد فلترتها

H: هو الفلتر المستخدم

مثال:

نأخذ صورة معينة ونقوم بإضافة ضجيج لها ثم نستخدم الفلتر Mean Filter لتحسين الصورة

```
i=imread('img1.jpg');
imshow(i);

n=imnoise(i,'gaussian');
imshow(n);

meanf1=fspecial('average',3);
fImg1=imfilter(n,meanf1);
imshow(fImg1);

meanf2=fspecial('average',7);
fImg2=imfilter(n,meanf2);

imshow(fImg2);
```

*قارن بين الصورتين الناتجتين؟

- ويمكن القيام بالفلترة Mean Filter من خلال الكود التالي:

```
function [img] = mean_filter(pic,n)
pic=rgb2gray(pic);
pic=im2double(pic);
[x,y]=size(pic);
b=[3 5 7 9];
t=find(n==b);
s=t+t;
u=t+1;
a=ones(x+s,y+s);
```

```
a(u:x+t,u:y+t)=pic(:,:);
a(1:t,u:y+t)=a(u:s,u:y+t);
a(x+u:x+s,u:y+t)=a(x+1:x+t,u:y+t);
 x+s,u:s);
a(1:x+s,y+u:y+s)=a(1:x+s,y+1:y+t);
img=a;
for i=u:x
    for j=u:y
        img(i,j)=mean(mean(a(i-t:i+t ,j-t:j+t)));
    end
end
```

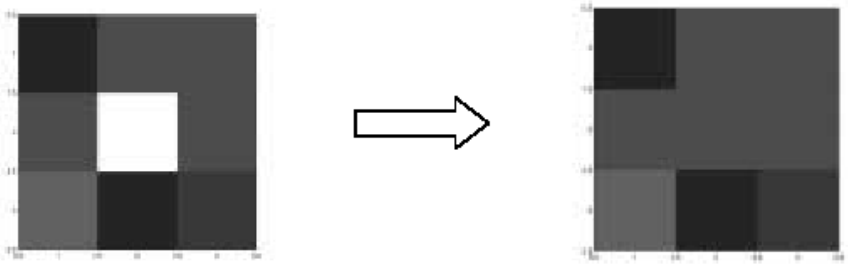
Median Filter -2

10	20	20
20	100	20
25	10	15

10,10,15,20,20,20,20,100]

median value

Therefore, replace 100 with 20



تطبق عملية الفلترة بواسطة الـ median filter من خلال التعليمة التالية:

B = medfilt2(A, [m n])

B = medfilt2(A)

A: تمثل الصورة المراد فلترتها

[m,n]: تمثل الأبعاد التي سيتم الترشيح وفقا لها (أي أبعاد مصفوفة الترشيح) وإذا لم يتم ذكرها فستكون الأبعاد [3,3]

مثال:

```
i=imread('img1.jpg');
imshow(i)

n=imnoise(i,'Gaussian');
figure,imshow(n)

medf1=medfilt2(n,[3 3]);

figure,imshow(medf1);

medf2=medfilt2(n,[5 5]);
figure,imshow(medf2);
```

*قارن بين الصورتين الناتجتين؟

- ويمكن القيام بالفلترة Median Filter من خلال الكود التالي:

```
function [img] = median_filter(pic,n)
pic=rgb2gray(pic);
pic=im2double(pic);
[x,y]=size(pic);
```

```
b=[3 5 7 9];
t=find(n==b);
s=t+t;
u=t+1;
a=ones(x+s,y+s);
a(u:x+t,u:y+t)=pic(:,:);
a(1:t,u:y+t)=a(u:s,u:y+t);
a(x+u:x+s,u:y+t)=a(x+1:x+t,u:y+t);
a(1:x+s,1:t)=a(1:x+s,u:s);
a(1:x+s,y+u:y+s)=a(1:x+s,y+1:y+t);
for i=u:x
    for j=u:y
        m=a(i-t:i+t,j-t:j+t);
        img(i,j)=median(m(:));
    end
end
```

الهيستوغرام (Histogram):

هو عبارة عن مخطط بياني يربط منحنى السويات الرمادية للصورة مع عدد البكسلات الموجودة عند كل سوية، إذاً يمكننا من خلاله معرفة إن كانت الصورة أقرب للأبيض أو للأسود و يمكن من خلاله أيضاً تحديد التباين (contrast) في الصورة.

```
i= imread('coins.png');
imhist(i)
```

تسوية الهيستوغرام (Histogram Equalization):

ويستخدم من أجل تحسين تباين الصورة :

```
h=histeq(i);
imshow(h)
imhist(h)
```

كشف الحواف Edge Detection :

اكتشاف الحواف هي عبارة عن ترشيح + تعتيب (thresholding)

وخرج عملية اكتشاف الحواف لصورة هي صورة الخطوط البيضاء فهي تعبر عن أمكنة وجود الحواف في الصورة الأصلية

وهناك طرق كثيرة لاكتشاف الحواف منها:1- sobel

2-prewitt

3- roberts

4- canny

ولكل طريقة قيمة تسمى thresholding تحدد العتبة التي تتم عندها عملية كشف الحواف، وذلك عن طريق مقارنة السوية الرمادية لكل بكسل مع هذه العتبة فإذا كانت هذه السوية أكبر من العتبة ستكون قيمة البكسل في الخرج واحد اي أبيض وإذا كانت اقل ستكون قيمة البكسل في الخرج صفر أي أسود وهذا يعني ان خرج عملية كشف الحواف صورة Binary

في حال عدم تحديد قيمة العتبة يأخذ التابع edge قيمة عتبة بشكل افتراضي

```
i= imread('coins.png');

BW=edge(i,'sobel');
```

و إذا أردنا معرفة قيمة العتبة التي تم تحديد الحواف عندها يمكننا عن طريق ما يلي:

```
[bw thresh]=edge(i,'sobel');
```

كما ويمكننا تحديد قيمة العتبة التي نريد كما يلي:

```
BW=edge(i,'sobel',0.21);
imshow(BW)
```

وفي بعض الأحيان يهمّنا تحديد اتجاه كشف الحواف فيمكن ان يكون شاقولي " vertical" أي يتم كشف الحواف فقط في الاتجاه الشاقولي، ويمكن أن يكون أفقي" horizontal" أي يتم كشف الحواف في الاتجاه الأفقي فقط

وفي حال عدم تحديد الاتجاه يتم كشف الحواف في الاتجاهين

```
BW=edge(i,'sobel',0.21,'vertical');
```

فيما يخص الطريقة prewitt نفس البارامترات:

```
i= imread('coins.png');
BW=edge(i,'prewitt');
BW=edge(i,'prewitt',0.21);
imshow(BW)
```

roberts:

```
BW=edge(i,'prewitt',0.21,'vertical');
i= imread('coins.png');
BW=edge(i,'roberts');
BW=edge(i,'roberts',0.21);
imshow(BW)
BW=edge(i,'roberts',0.21);
```

أمّا الطريقة الأفضل على الإطلاق في مجال كشف الحواف هي canny:

وذلك لأنّ كشف الحواف فيها يعتمد على شعاع [low high] الحواف الناتجة عن العتبة العليا تسمى strong edges والحواف الناتجة عن العتبة الدنيا تسمى weak edges ، يتم في البداية كشف الحواف القوية ومن ثم كشف الحواف الضعيفة المرتبطة مع الحواف القوية .

```
i= imread('coins.png');
BW=edge(i,'canny');
imshow(BW)
 [bw thresh]=edge(i,'canny');
```

ملاحظة: low= 0.4*high

هذا البرنامج يبين تأثير الضجيج على عملية كشف الحواف:

```
i= imread('coins.png');
n=imnoise(i,'salt & pepper');
imshow(n)
BW=edge(n,'sobel');
figure,imshow(BW)
BW=edge(n,'prewitt');
figure,imshow(BW)
BW=edge(n,'roberts');
figure,imshow(BW)
BW=edge(n,'canny');
figure,imshow(BW)
```

بوجود الضجيج تسوء عملية كشف الحواف للغاية ولذلك إن الحل الأمثل لكشف الحواف في حال وجود ضجيج هو عملية الفلترة ومن ثم كشف الحواف:

```
i= imread('coins.png');
n=imnoise(i,'salt & pepper');
h=fspecial('average');
o=imfilter(n,h);
figure, imshow(o)
BW=edge(o,'sobel');
figure,imshow(BW)
BW=edge(o,'prewitt');
figure,imshow(BW)
BW=edge(o,'roberts');
figure,imshow(BW)
BW=edge(o,'canny');
figure,imshow(BW)
```

وظيفة:

لدينا صورة pic4 ذات امتداد png اذا اسندناها إلى متحول i سيكون من النوع uint8 والمطلوب إيجاد الصور الثمانية الناتجة عن تأثير كل بت على حدا.