NEIGHBOURHOOD PROCESSING (KOMŞULUK İLİŞKİLİ İŞLEMLERİ- BÖLGESEL İŞLEMLER-UZAYSAL FİLTRELEME)

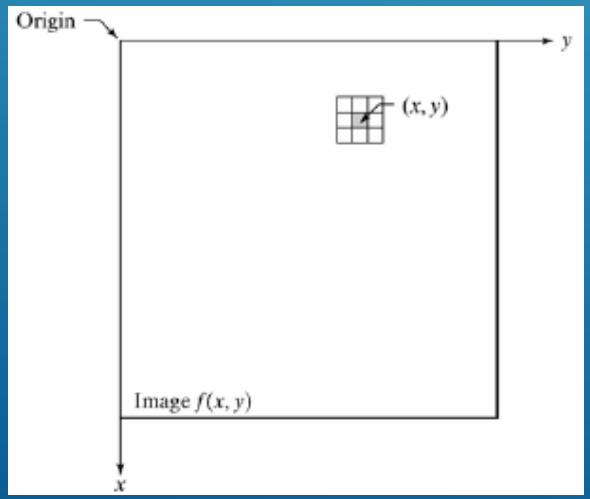
KOMŞULUK İLİŞKİLİ İŞLEMLER (UZAYSAL FİLİTRELER)

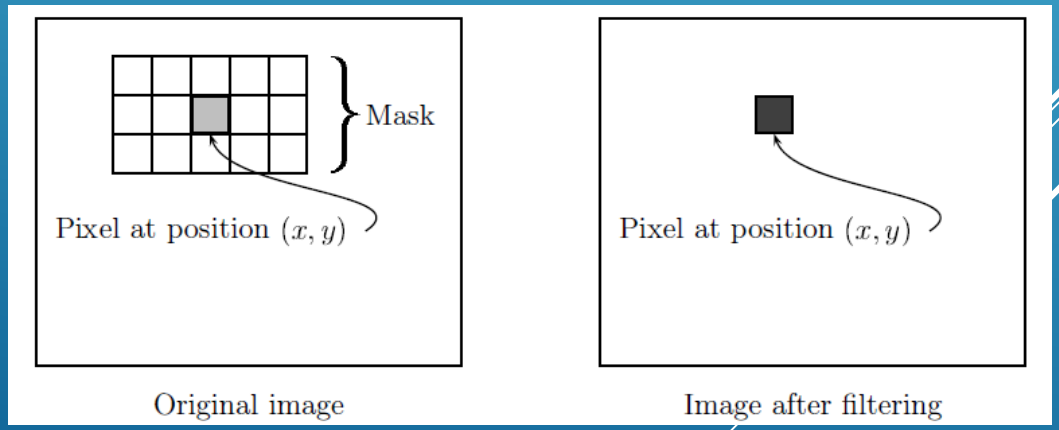
Noktasal işlemler imgedeki her piksele, diğer piksellerden bağımsız olarak y=f(x) işleminin uygulanışıydı.

Komşuluk ilişkili işlemler (Bölgesel İşlemler) ise; Noktasal işlemlerin genişletilerek y=f(x) fonksiyonunu her bir pikselin komşuluk ilişkilerine göre merkez piksele uygulamaktır. Noktasal işlemlere göre aradaki fark; her bir piksele bağımsız olarak işlem yapılması yerine, merkez pikselin yeni değeri hesaplanırken komşu piksellerinde göz önüne alınmasıdır;

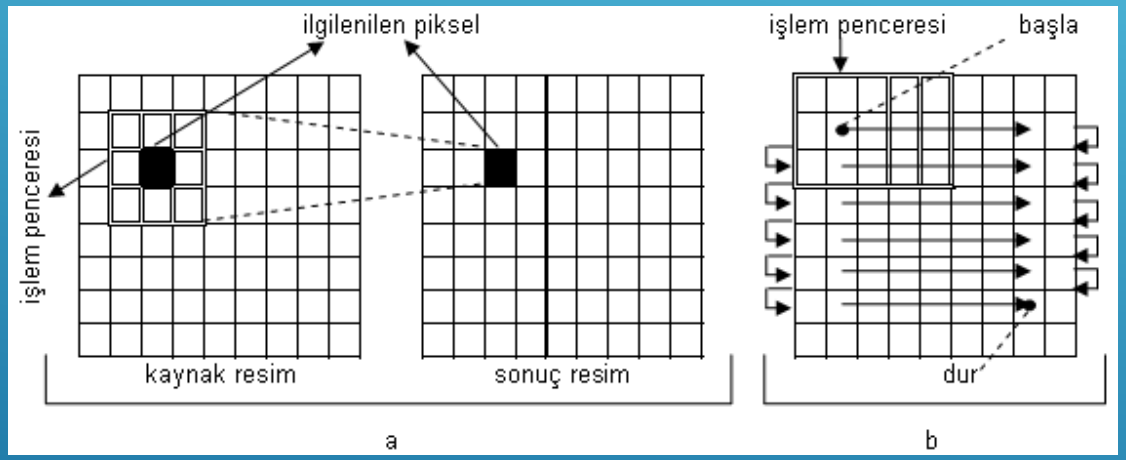
Komşuluk ilişkili işlemler aşağıda tanımlanan 4 adımdan oluşur.

- Bir merkez nokta tanımlamak (x,y)
- Bu merkez noktasının yeni değerini elde etmek için, sadece önceden tanımlanmış komşuların piksellerini de içeren bir operasyon yapmak.
- O noktadaki (Merkez noktadaki) işlemin o operasyonun cevabı olmasına izin vermek;
- İmgedeki her nokta için bu işlemi tekrarlamak (Merkez noktayı kaydırma).





- Bölgesel işlemlerin amacı komşu piksellerin gri tonlarını vurgulamak veya "görünmesine engel olmaktır. Bu işlem şekilde görülmektedir.
- ***Merkezi piksel etrafında tanımlanan komşuluğa "maske" veya " pencere"*** denir.
- Bütün resmin işlenmesi için, maskenin adım adım kaydırılarak resmin tamamını taraması gerekir (Bu işlem Merkez noktayı kaydırma işlemidir. Yani bir görüntüdeki herbir piksel için bir tane olacak şekilde yeni komşuluklar oluşturma işlemidir.). Bu işlem resmin sol üst köşesinden başlar . Yeni griton hesaplandıktan sonra maske (pencere) bir piksel sağ tarafa kaldırılır. Yeniden hesaplanır ve aynı işlemlere devam edilir. Satırın sonuna gelince, işlemleri bir sonraki satırın başından devam edilir.



a)Belirli bir algoritma ile pencere içerisindeki gritonlar işleme tabi tutulur ve sonuç resimde aynı pozisyona atanır.
b)pencere adım adım kaydırılarak resmin tamamı taranır.

- İlgilenilen pikselin resmin sınırlarına ulaşmadığı açıkça gözükmektedir. Bu yüzden resim bölgesel işlemlerden dolayı küçülür. Genellikle bu küçülme önemli değildir. Fakat kenar piksellere griton verilmediğinden emin olunması gerekir. Böyle problemlere engel olmak için başlangıçta sonuç resmin bütün piksellerine 0 değeri atanmalıdır.

KOMŞULUK İLİŞKİLİ İŞLEMLER (2)

Bölgesel resim işlemenin iki önemli kuralı;

1-Sonuç resim, kaynak resimden ayırt edilmeli.

2-Bir işleme başlamadan önce sonuç resmin bütün değerlerine 0 atanmalı.

- Komşuluk ilişkili işlemler için en uygun yol; uygun bir maske (filtre maskesi-pencere) oluşturup bu maskeyi tüm görüntü üzerinde hareket ettirerek y=f(x) fonksiyonunu uygulamaktır. (Bu maske genellikle, kenarları tek sayıda piksellerden oluşturulmuş dikdörtgen şekildir.)

- Bu maske ve fonksiyonun kombinasyonuna Filtreleme denir. (Filtreleme işlemi çoğu literatürde komşuluk ilişkili İşlemlerin yerine kullanılır)

- Eğer pikselde yeni gri seviye oluşturma fonksiyonu lineer ise ve maskedeki tüm gri seviye değerleri için lineer uygulanıyorsa bu filtre , Lineer Uzaysal Filtre (Linear Spatial Filter) 'dir.

- Üç klasik bölgesel filtreleme vardır.

  1- **Griton (Gri seviye) düzgünleştirme** (grey level smoothing - Yumuşatma),

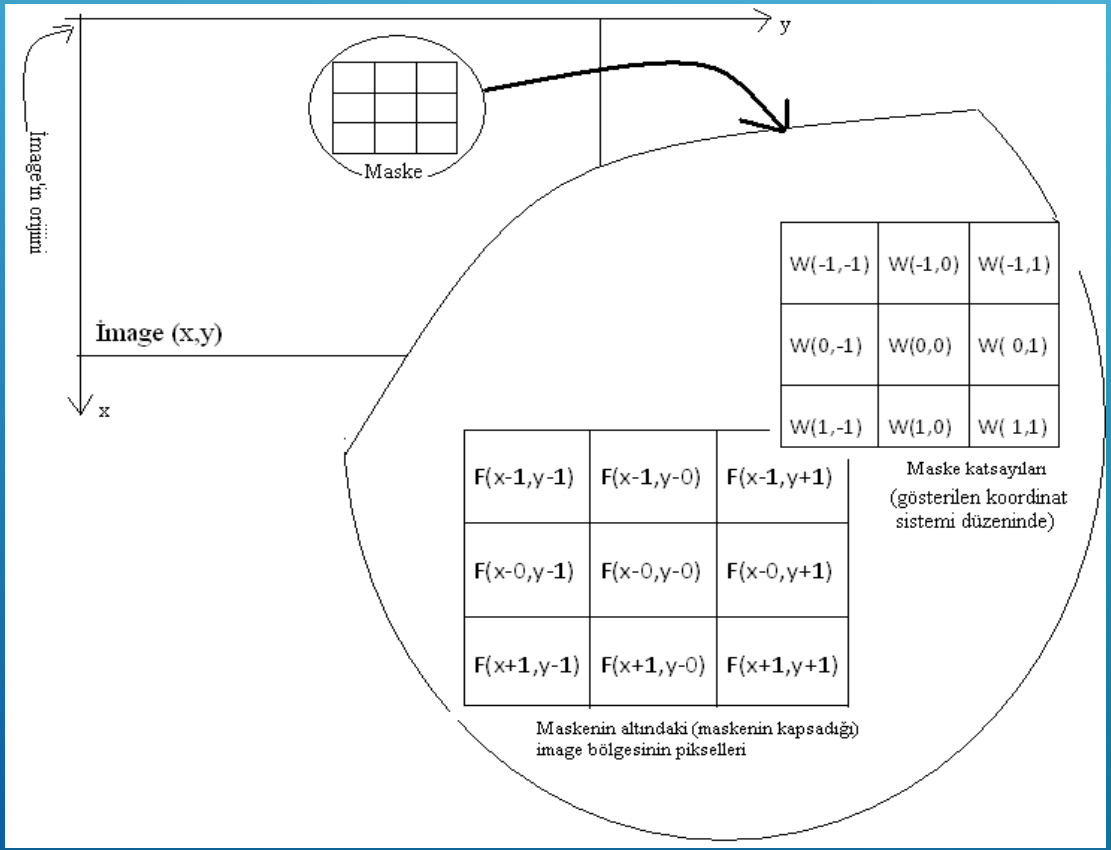
  2- Griton farklılıklarını vurgulamak (emphasizing grey level differences)

  3- **Griton geçişlerini keskinleştirmek** (sharpening grey level steps - Keskinleştirme).

LİNEER UZAYSAL FİLTRENİN YAPISI

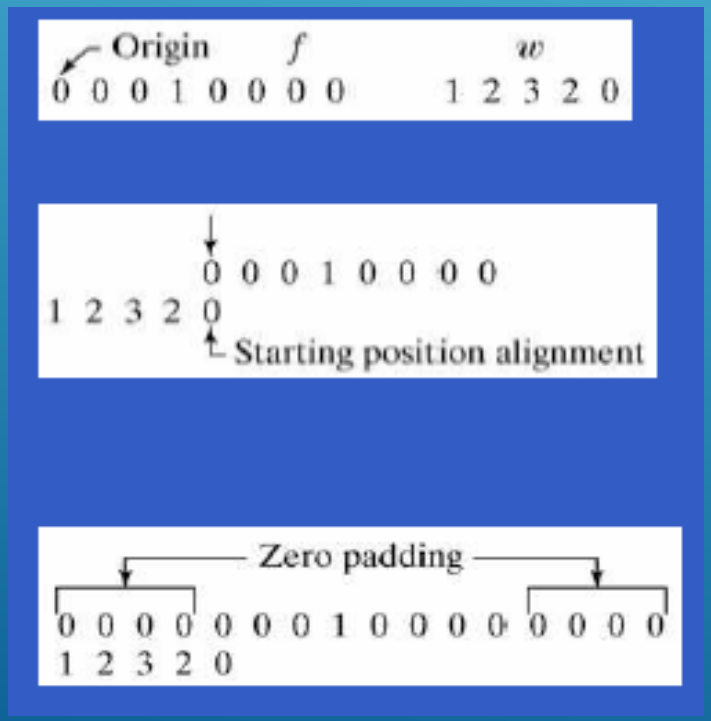
3x3'lük bir maske ve onunla ilgili komşuluk ilişkili bölge altındaki pikseller resimde görülmektedir. Lineer uzaysal filtrelemenin daha iyi açıklanabilmesi için iki önemli kavram vardır. Korelasyon ve Konvolüsyon.

- Korelasyon; maske w'nin f görüntü dizisinden geçirilme işlemidir.
- Konvolusyon ; maskenin 180 ◦ döndürülüp aynı işlemin yapılma sürecidir.



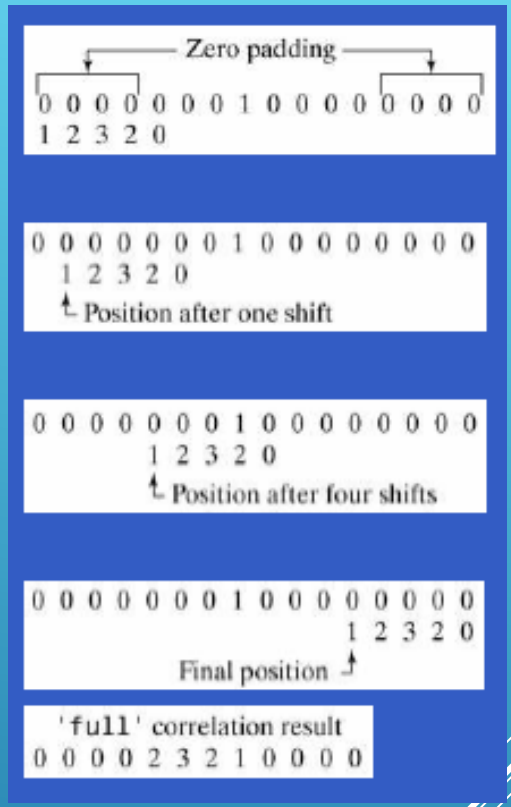
TEK BOYUTLU KORELASYON

- Resimde f dizisinin, w dizisi ile korelasyon işlemi görülmektedir. İki fonksiyonun korelasyonunu başarmak için;
- w fonksiyonunun en sağdaki noktasının ,f fonksiyonun en soldaki noktasına gelecek şekilde konumlandırarak işlem başlatılıp, her seferinde w dizisini bir adım sağa kaydırarak işlemler tekrarlanır.
- F dizisi ile w dizisinin elemanları üste örtüşmeyebilir. Bu durumda f dizisinin sağına ve soluna gerektiği kadar 0 doldurulur.
- Böylelikle f dizisinin her elemanının w dizisi ile ilişkiye girmesi garanti edilmiş olur.



TEK BOYUTLU KORELASYON (2)

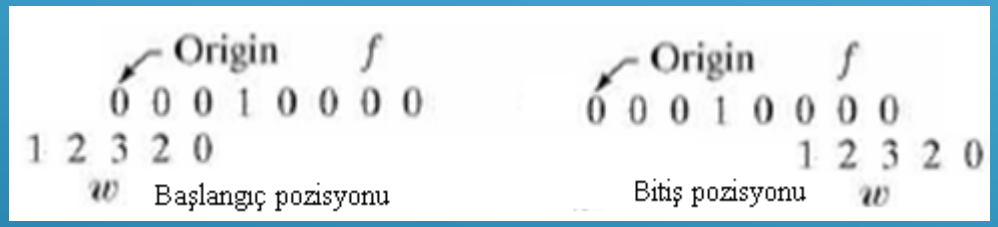
- Korelasyonun ilk değeri, 1.pozisyondaki her iki fonksiyonun eleman-eleman çarpımlarının toplamıdır. (0x1+0x2+0x3+0x2+0x0=0)
- İkinci değeri bulmak için; w dizisi 1 adım sağa kaydırılarak tekrar eleman-eleman çarpılıp toplanır. (0x1+0x2+0x3+0x2+0x0=0).
- w bir defa daha sağa kaydırılıp aynı işlemler tekrarlanarak yeni dizinin 3.elemanı da 0 olarak bulunur.
- w 3.kez sağa kaydırılıp aynı işlemler yapılırsa yeni dizinin 4.elemanıda 0 çıkar.
- w 4.kez kaydırılıp aynı işlemler yapılırsa dizinin 5. elemanı 2 çıkar (0x1+2x0+3x0+2x1+0x0=2). Bu korelasyonun 0'dan farklı ilk değeridir.
- w'yı kaydırıp çarpımların toplamı işlemine devam edilierek final posisyonuna gelinir.



FULL etiketi, korelasyon işleminin, f dizisinin sağına ve soluna dolgu eklenerek yapıldığının ve bütün değerler için hesaplandığını gösterir.

TEK BOYUTLU KORELASYON (3)

- Alternatif bir opsiyon ise (Same) ; korelasyonla elde edilen yeni dizinin uzunluğu f dizisi ile aynıdır. Bu hesaplama da sıfır dolgularını kullanır. Fakat başlama pozisyonunda; maskenin orta noktası (bu maskede 3) ile f dizisinin orijini (dikkat f dizisinin orijini en sol basamak) aynı hizadadır. En son hesaplama ise maskenin merkez noktasının f'nin son noktasına hizalandığı pozisyondur. (Buna göre aşağıdaki örneği inceleyiniz.)



'full' correlation result
0 0 0 0 2 3 2 1 0 0 0 0

'same' correlation result
0 0 2 3 2 1 0 0

SPATİAL CORRELATİON (UZAYSAL KORELASYON)

mxn boyutlu bir w(x,y) filtresi ile bir f(x,y) görüntüsünün korelasyonu;
W (x,y) ☆ f(x,y) şeklinde gösterilir.

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

2 BOYUTLU KORELASYON

Tek boyutlu korelasyon görüntüye kolaylıkla adapte edilebilir.

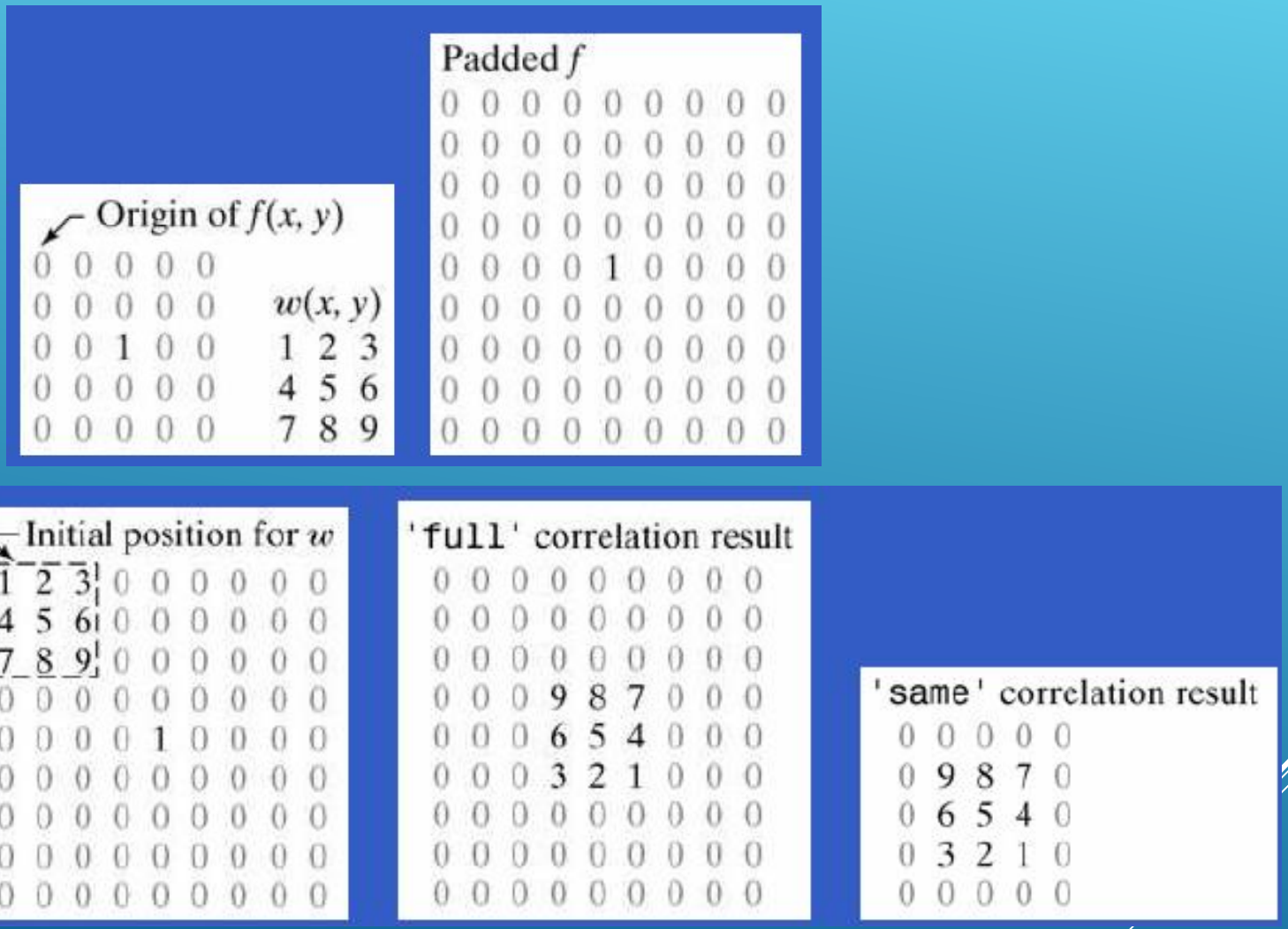


İMAGE FİLTRELEME

Filtreleme resmin üzerinde bir filtre varmış gibi düşünüp her piksel değerinin yeniden hesaplanmasıdır. Filtreler sayesinde girdi resminden yeni resim değişik efektler verilerek elde edilir.

Filtreleme işlemi aşağıdaki formülle elde edilir. Burada m (s,t) filtre matrisidir (ağırlıklar matrisi-Kernel - Maske) . P (i,j) filtrelenecek görüntünün matrisidir. Y(i,j) elde edilecek yeni görüntünün matrisidir (Bu formül 2 boyutlu korelasyon formülü gibi yorumlanabilir).

$$y(i,j) = \sum_{s=-1}^{1} \sum_{t=-2}^{2} m(s,t)p(i+s, j+t).$$

Burada filtre matrisimzi 3 x5 boyutunda bir matris olup, merkez pikseli, yeni değeri hesaplanacak pikselin üzerine gelecek şekilde konumlandırılıp, denkleme göre yeni değerler hesaplanır. Maske p(i,j) matrisinin tüm elemanlarını kapsayacak şekilde gezdirilerek, yeni piksel değeri kendisinin ve komşu elemanların oluşturacağı ağırlıklar toplamı şeklinde bulunur.

$m(-1,-2)$	$m(-1,-1)$	$m(-1,0)$	$m(-1,1)$	$m(-1,2)$
$m(0,-2)$	$m(0,-1)$	$m(0,0)$	$m(0,1)$	$m(0,2)$
$m(1,-2)$	$m(1,-1)$	$m(1,0)$	$m(1,1)$	$m(1,2)$

3 x 5 Ağırlıklar (Maske - kernel-Filtre) matrisi

$p(i-1,j-2)$	$p(i-1,j-1)$	$p(i-1,j)$	$p(i-1,j+1)$	$p(i-1,j+2)$
$p(i,j-2)$	$p(i,j-1)$	$p(i,j)$	$p(i,j+1)$	$p(i,j+2)$
$p(i+1,j-2)$	$p(i+1,j-1)$	$p(i+1,j)$	$p(i+1,j+1)$	$p(i+1,j+2)$

Filtrelenecek görüntü matrisinin bir kısmı

Uzaysal Filtre işlemini gerçekleştirmek için 3 adım gerekir.

1-Maskeyi (Filtreyi) , çalışılacak piksel merkeze gelecek şekilde konumlamak.

2- Maskenin tüm elemanları ile ilgili pikselle komşuluk ilişkisi olan piksellerin çarpılması

3-çarpımların toplanması işlemi.

Bu işlem tüm imgedeki tüm pikseller için gerçekleştirilir.

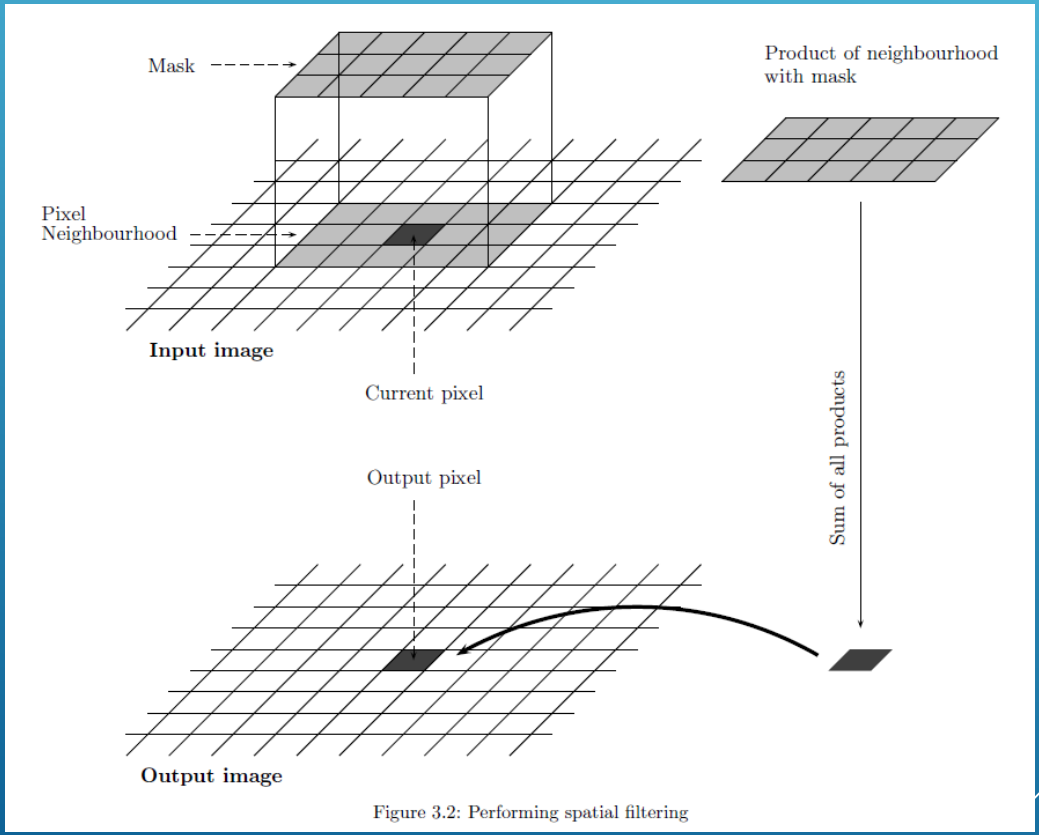


Figure 3.2: Performing spatial filtering

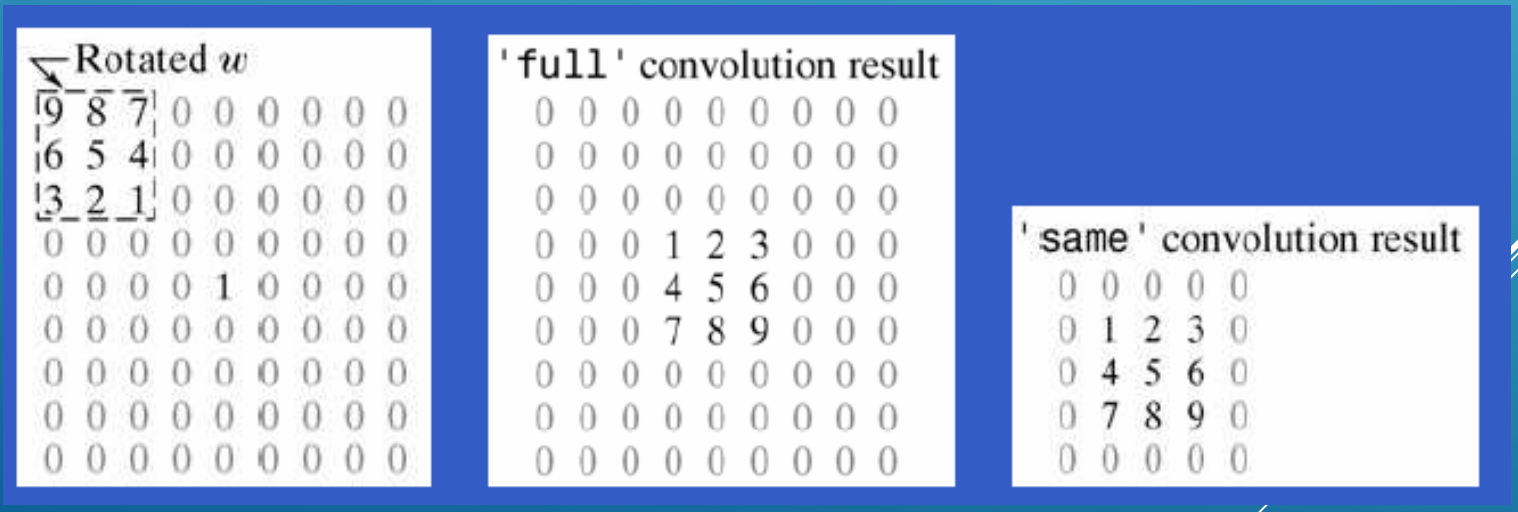
UZAYSAL KONVOLUSYON

Yukarıda analtılan genel uzaysal filtrelemenin önemli özel bir hali ise uzaysal konvolusyondur.

Korelasyona göre tek farkı, maske (filtre) matrisinin aşağıdan yukarı ve sağdan sola 180 derece dönüştürülmüş haliyle aynı işlemleri yapmaktır.

Bir görüntünün uzaysal filtrelenmesi konvolüsyon olarak adlandırılan bir işlemle yapılır. Konvolüsyonda bir pikselin çıkış değeri kendisinin ve komşu piksellerin değerlerinin bir ağırlıklı toplamı olarak bulunur.

Ağırlıklar matrisi konvolüsyon kerneli, maske, şablon veya impuls yanıtı olarak adlandırılır.



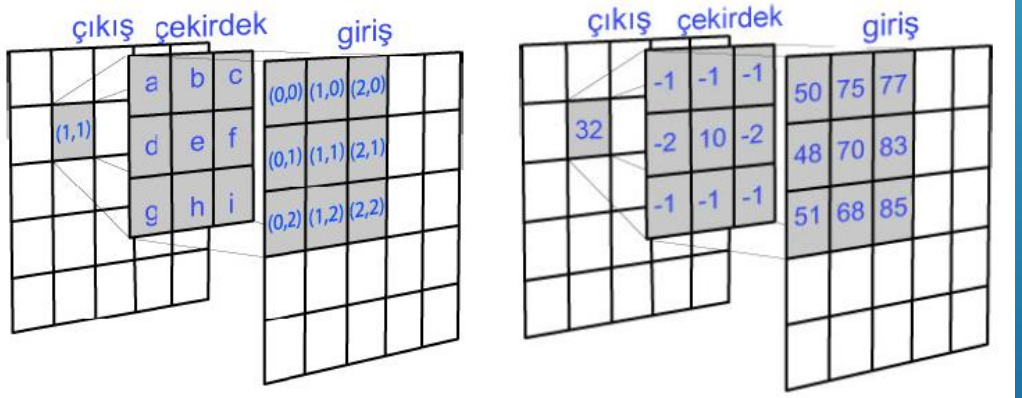
Aşağıdaki denklem iki boyutlu konvolüsyonu ifade ederler. Buna göre *f(x,y)* orijinal image, *h(i,j)* ağırlık (filtre-Maske-İmpulse yanıtı) matrisidir. *f'* (x,y) ise elde edilen yeni imagedir.

$$f'(x,y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j) \times f(x-i, y-j)$$

- Konvolüsyon; yumuşatma, keskinleştirme, kenar belirleme gibi görüntü işleme fonksiyonlarını gerçekleştirmede çok sık kullanılmaktadır.
- Konvolüsyonda bir pikselin yeni değeri kendisinin ve çevresindeki piksellerin ağırlıklı ortalaması ile bulunmaktadır. Konvolüsyon şablonu (Filtre matrisi-kernel) uygulamaya göre farklı boyutlarda olabilmekle beraber genelde 3x3 lük bir matristir.

$$imge = \begin{bmatrix} i(0,0) & i(0,1) & i(0,2) \\ i(1,0) & i(1,1) & i(1,2) \\ i(2,0) & i(2,1) & i(2,2) \end{bmatrix} = \begin{bmatrix} 55 & 75 & 77 \\ 48 & 70 & 83 \\ 51 & 68 & 85 \end{bmatrix} \quad çekirdek = \begin{bmatrix} -1 & -1 & -1 \\ -2 & 10 & -2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$i(1,1) = (-1x55) + (-1x75) + (-1x77) + (-2x48) + (10x70) + (-2x83) + (-1x51) + (-1x68) + (-1x85) = 32 \qquad (3.4)$$



Örnek: A giriş görüntüsünün h kerneli ile konvolüsyonu sonucu oluşan çıkış görüntüsünün (2,4) pikselinin değerini hesaplayalım:

Çözüm:

1. h kernelini yatay ve düşey eksende 180 derece döndür.
2. h'ın merkez elemanı A(2,4) noktasına çakışacak şekilde kerneli kaydır.
3. A ve h'daki karşılıklı elemanları çarp ve hepsini toplayarak (2,4) pikselindeki çıkış değerini bul.

$$A = \begin{vmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{vmatrix}$$

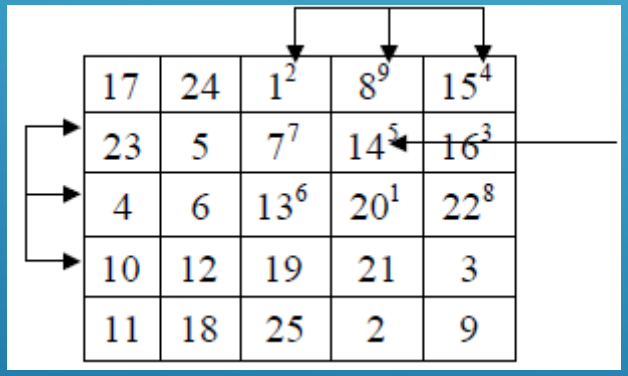
$$h = \begin{vmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{vmatrix}$$

ÖRNEK DEVAM

$$A=\begin{vmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{vmatrix} \qquad h=\begin{vmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{vmatrix}$$

Döndürülmüş kernel değerleri

Giriş görüntüsünün değerleri



Çıkışını bulmak istediğimiz piksel

(2,4) pikselinin çıkışı

$$1.2+8.9+15.4+7.7+14.5+16.3+13.6+20.1+22.8=575$$

BAZI ÖNEMLİ LİNEER UZAYSAL FİLTRELER

Gaussian Blur, bulanıklaştırma derecesini ayarlamamıza olanak vererek görüntünün hafifçe yumuşatılmasından tüm görüntüyü kalın bir sisle kaplamaya kadar değişen efektler yaratır. Adını, renk değerlerinin değişimini gaussian çanı denilen eğriyle eşleştirmesinden alır.

Basic 3x3 blurring filter

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = 1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Gaussian 3x3 blurring filter

$$1/16 \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Gaussian 5x5 blurring filter

$$1/112 \times \begin{bmatrix} 1 & 2 & 4 & 2 & 1 \\ 2 & 6 & 9 & 6 & 2 \\ 4 & 9 & 16 & 9 & 4 \\ 2 & 6 & 9 & 6 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{bmatrix}$$

Horizontal **Vertical** **Diagonal**

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

Prewitt filters

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Roberts filters

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

Sobel filters

GRİTON DÜZGÜNLEŞTİRME (GREY LEVEL SMOOTHING) LİNEER FİLTRELERİ NORMAL ORTALAMA (TEMEL BLURRİNG (BULANIKLAŞTIRMA)) FİLTRE

Aşağıdaki kaynak resim iki bölgeden oluşmaktadır: Karanlık kısım (griton 1) ve aydınlık kısım (griton 10). Diğer gritonlar gürültü olarak yorumlanırsa yapılacak iş ya onları yok etmek veya diğer bir deyişle iki bölgeyi elde etmektir. En basit düzgünleştirme metodu ortalama işlemidir. kaynak resme ortalama işlemi uygulanarak elde edilen sonuç resmi yanda görülmektedir.

Uygulanan maskenin boyutları 3x3'tür. Maske içerisindeki piksellerin gritonları toplanıp 9'a bölünür. Açıkça görüleceği üzere gürültü pikselleri gritonları arzulanan griton değerlerine yaklaştırıldı. Buna rağmen kaynak resimdeki bölgeler arasındaki dik olan geçişi yassılaştı (bulanıklaştı). Pozitif veya negatif etki olarak bunun kıymeti uygulamaya bağlıdır.

1x1+1x1+1x9+1x1+2x1+8x1+1x1+1x1+10x1 = 35 /9 = 3.8 = 4

1	1	1	1	10	10	10	10
1	1	6	1	8	10	2	10
1	3	1	1	9	10	7	10
1	1	1	2	8	9	10	10
1	1	1	1	10	10	10	10
1	4	1	2	9	10	2	10
1	2	1	8	10	10	10	10
1	1	1	1	10	10	10	10

Kaynak resim

0	0	0	0	0	0	0	0
0	2	2	4	7	8	9	0
0	2	2	4	6	8	9	0
0	1	1	4	7	9	10	0
0	1	2	4	7	9	9	0
0	1	2	5	8	9	9	0
0	1	2	5	8	9	9	0
0	0	0	0	0	0	0	0

kaynak resme 3x3 ortalama maskesi kullanılarak elde edilen sonuç resim

GAUSSİAN BLURRİNG FİLTRE (AĞIRLIKLI ORTALAMA)

Bu durumda, maske içerisinde ki gri tonlar belirli ağırlıklarla (coefficients (kat sayısı)) çarpılır. Şekilde "Gausian alçak geçiren filtre" ve normal ortalamanın ağırlıklarını gösterilmektedir. Normal ortalama operatörü (3x3 maske) durumunda, maske içerisindeki gritonlar eşit ağırlıklıdır (yani ağırlıklar 1 değerine eşittir). Maskenin şeklinden dolayı bu maskeyi kullanan filitreye kutu filitre denir.

Gaussian alçak geçirenin düzgünleştirme etkisi kutu filtreninkinden biraz daha iyidir. Fakat griton basamaklarının yassılaştırma problemi bunda da mevcuttur.

1/9

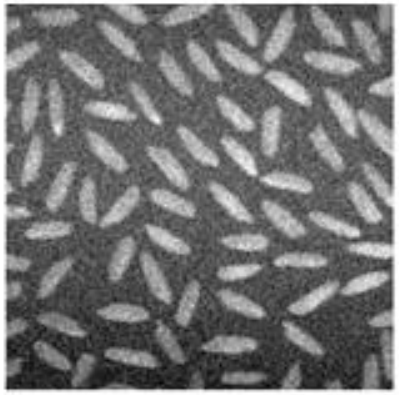
1	1	1
1	1	1
1	1	1

Normal Filtre (kutu filtre)

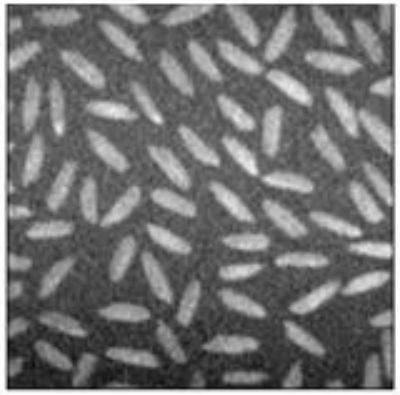
1/16

1	2	1
2	4	2
1	2	1

Gaussian alçak geçiren filtre



Gürültülü resim,



Gaussian alçak geçiren filtre uygulanmış sonuç resim

LİNEEER UZAYSAL FİLTRELER İÇİN MATLAB FONKSİYONLARI

Burada f giriş görüntüsü, w filtre maskesi(kernel), g filtrelenmiş sonuçtur. Diğer parametreler ise tabloda gösterilmektedir.

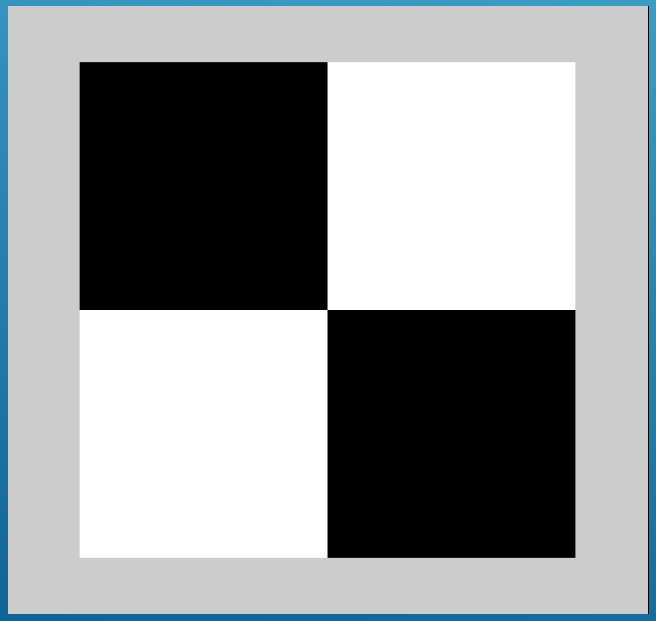
```
g=imfilter(f,w,filtering_mode,...
           boundary_options,size_options)
```

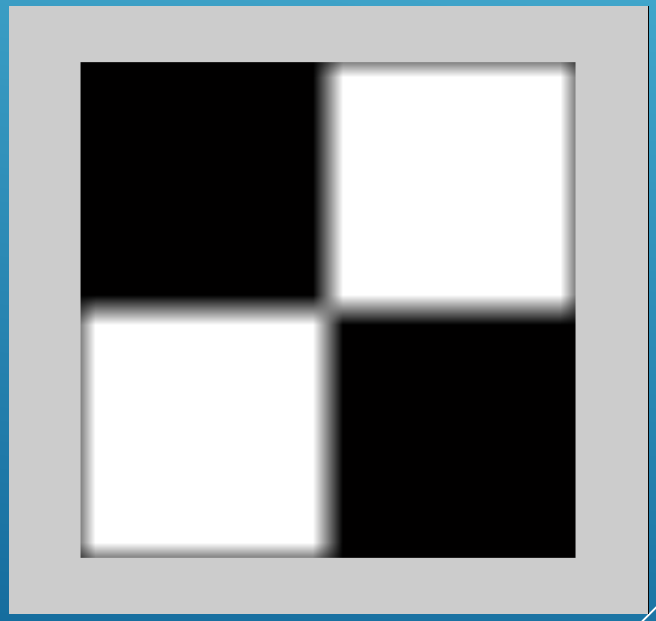
Options	Description
Filtering Mode	
'corr'	Filtering is done using correlation. This is the default.
'conv'	Filtering is done using convolution.
Boundary Options	
P	The boundaries of the input image are extended by padding with a value, P. This is the default, with value 0.
'replicate'	The size of the image is extended by replicating the values in its outer border.
'symmetric'	The size of the image is extended by mirror-reflecting it across its border.
'circular'	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
'full'	The output is of the same size as the extended (padded) image.
'same'	The output is of the same size as the input. This is the default.

ÖRNEK: ORTALAMA FİLTRE

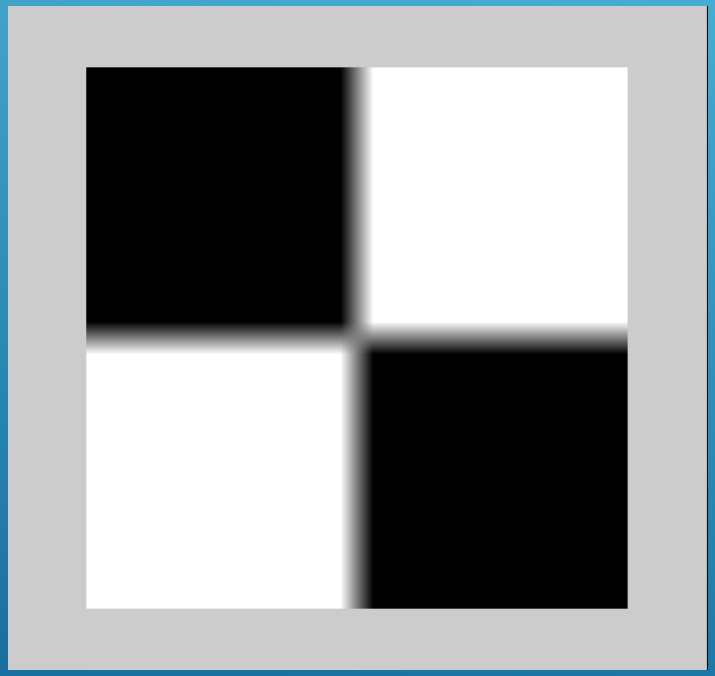
```
>> X=[ZEROS(256,256) ONES(256,256);ONES(256,256) ZEROS(256,256)];
>> Z=DOUBLE(X);
>> İMSHOW(Z)
```

```
>> X=[ZEROS(256,256) ONES(256,256);ONES(256,256) ZEROS(256,256)];
>> Z=DOUBLE(X);
>> W=ONES(31);
>> GD=İMFİLTER(Z,W);
>> İMSHOW(GD,[])
```

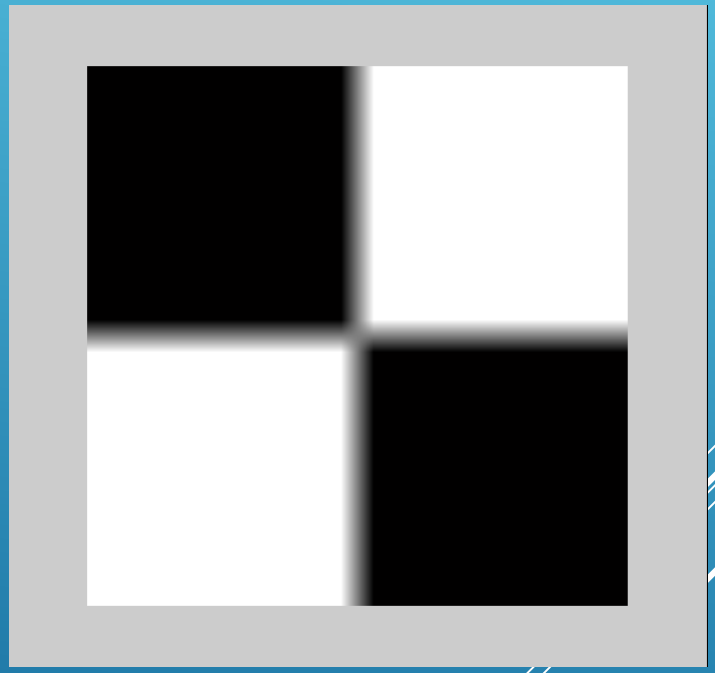




```
>> GR=İMFİLTER(Z,W,'REPLİCATE');
>> İMSHOW(GR,[])
```



```
>> GS=İMFİLTER(Z,W,'SYMMETRİC');
>> İMSHOW(GS,[])
```



ÖNEMLİ NOT-1

- 2 boyutlu görüntülerin filtrelenmesi için diğer bir fonksiyon ise filter2 fonksiyonudur. Bu fonksiyon aşağıdaki gibi kullanılır. Sonuç matrisinın veri tipi double'dır.

```
>> c = filter2(filter, image, shape)
```

- Filter parametresi; istenen filtrelemeyi (filtre tipini),
- İmage parametresi; filtrelenecek image'i,
- Shape parametresi;(same, full,valid olabilir.)
- same (filtrelenmiş görüntü orijinal görüntü boyutundadır)
- Full (Filtrelenmiş görüntü paddingden dolayı orijinal imajdan daha büyük boyutludur.)
- valid (Filtrelenmiş görüntü orijinal görüntüden küçüktür, Çünkü işlemlerden kenar pikseller etkilenmez).

DOĞRUSAL OLMAYAN UZAYSAL FİLTRELER (NONLİNEER SPATİAL FİLTERS)

- Komşuluk ilşikili Doğrusal olmayan filtreleme operasyonları m × n boyutlu filtre matrisinin merkez noktasının kaydırılması işlemidir. Bu haliyle komşuluk ilişkili doğrusal uzaysal filtreleme ile aynıdır.
- Doğrusal olmamanın anlamı her merkez nokta pikseline aynı işlemin uygulanmamasından kaynaklanır. Örneğin; herhangi bir merkez nokta pikselini komşu piksellerdeki en büyük piksel değerine eşitleme işlemi bir nonlineer filtreleme işlemidir.
- Daha az kullanılan bir tarif ise; maskenin nonlineer olmasıdır.
- Esas olan merkez piksellerinyeni değerlerinin bulunması için nonlineer işlemlerin yapılmasıdır.

MİNİMİZE OPERATÖRÜ (MİN OPERATÖRÜ) (NON LİNEER FİLTRELEMEDİR)

Griton basamaklarını koruyan çok basit düzgünleştirme operatörü ***"min operatörü"*** dür. ***min operatörü maske içerisindeki minimum gritonu yeni griton olarak verir.***

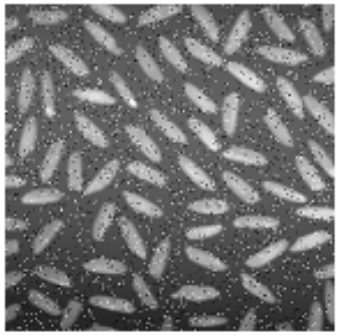
Şekillerden de görüleceği üzere resmin karanlık bölgesi (griton 1) temizlenir. fakat önceki aydınlık bölgede bozulmalar meydana gelir.

orijinal resim

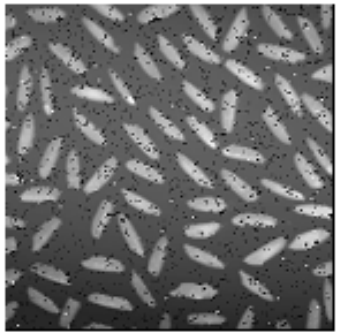
1	1	1	1	10	10	10	10
1	1	6	1	8	10	2	10
1	3	1	1	9	10	7	10
1	1	1	2	8	9	10	10
1	1	1	1	10	10	10	10
1	4	1	2	9	10	2	10
1	2	1	8	10	10	10	10
1	1	1	1	10	10	10	10

min operatörü ile elde edilen resim

0	0	0	0	0	0	0	0
0	1	1	1	1	2	2	0
0	1	1	1	1	2	2	0
0	1	1	1	1	8	7	0
0	1	1	1	1	2	2	0
0	1	1	1	1	2	2	0
0	1	1	1	1	2	2	0
0	0	0	0	0	0	0	0



Kaynak resim

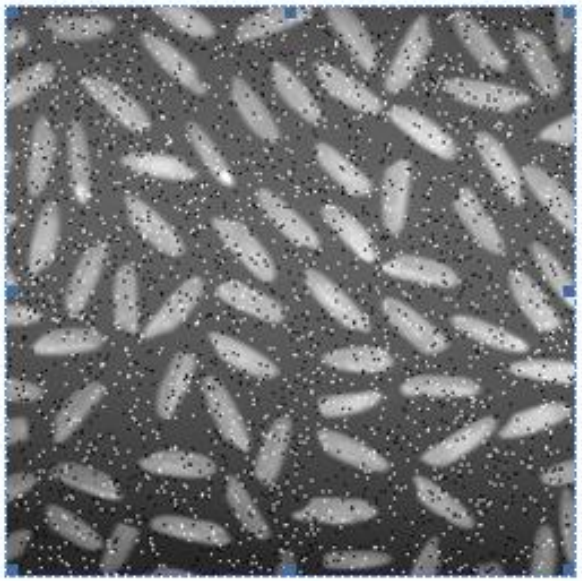


Min operatörü uygulanmış sonuç resim

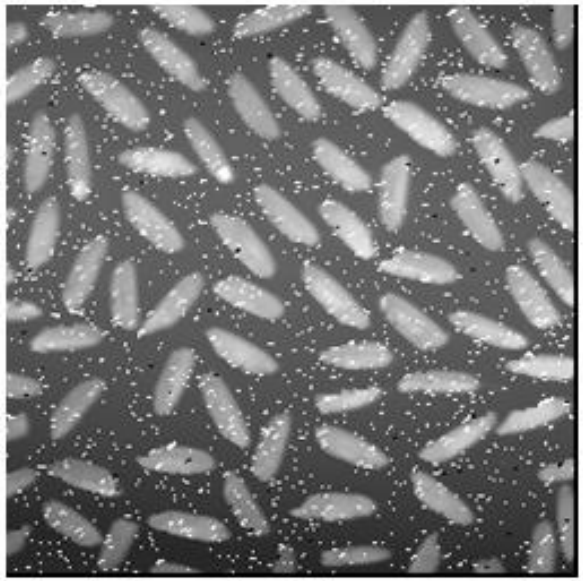
MAX OPERATÖRÜ: (BASİT BİR NONLİNEER FİLTRELEMEDİR)

Min operatörünü tamamlayıcı ***"max operatörü"dür***; ***yani maske içerisindeki en yüksek griton değeri, yeni griton değeri olarak sonuç resimde ilgilenilen noktaya atanır.***

Dolayısıyla sonuç resimde aydınlık bölge temizlenir. Fakat karanlık bölgede bozulmalar meydana gelir. Şekilde Max operatörünün resim üzerindeki etkisini göstermektedir.

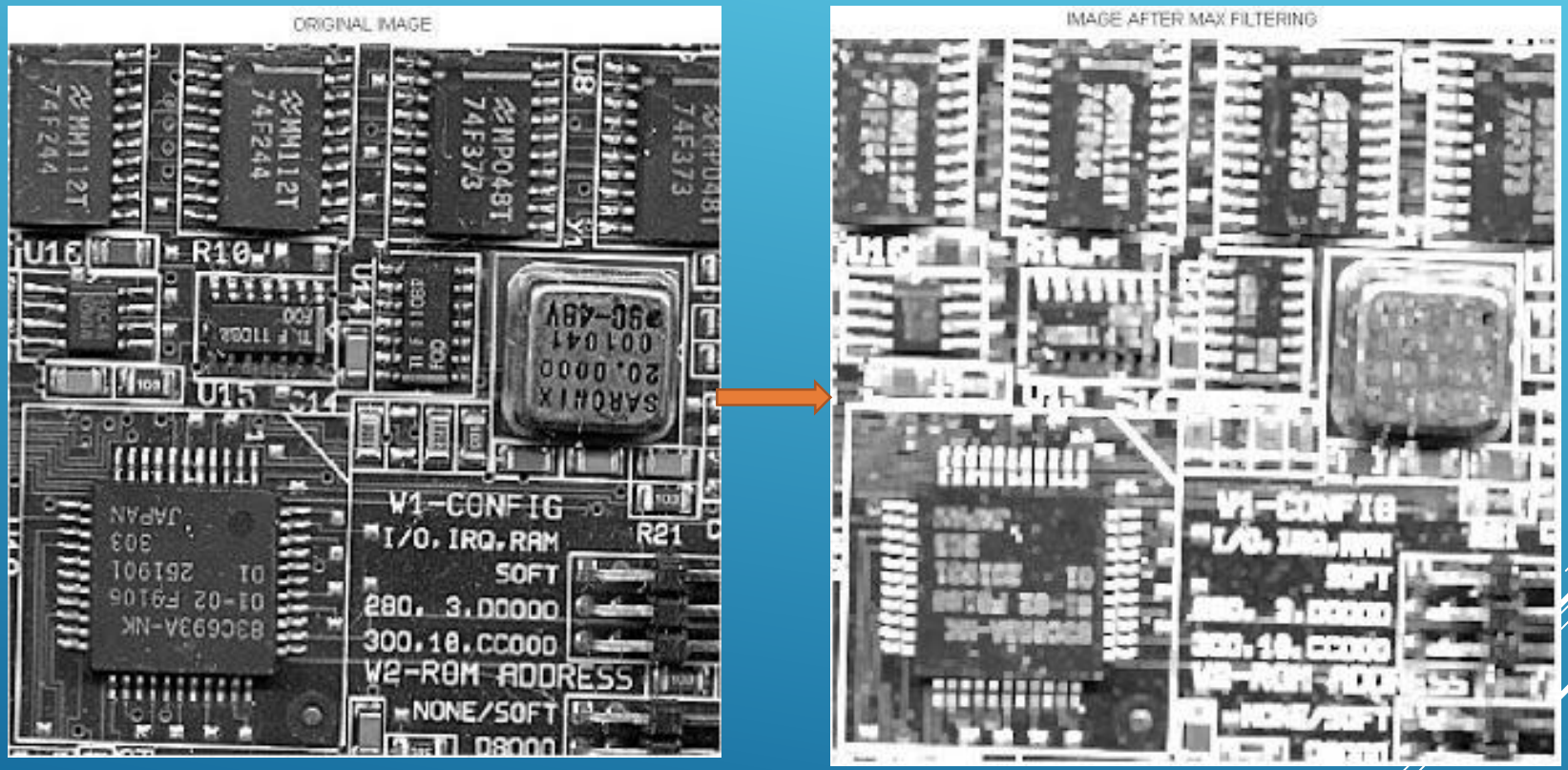


Kaynak resim.



Max operatörü uygulanmış sonuç resim

Max Filtre: Aşağıdaki imgeye 3x3 pencere boyutunda max filtre uygulayarak ikinci imgeyi elde eden matlab kodlarını yazalım.

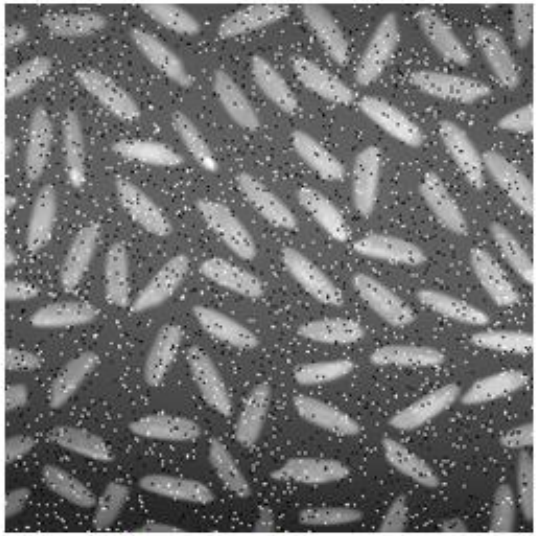


Max Filtre:

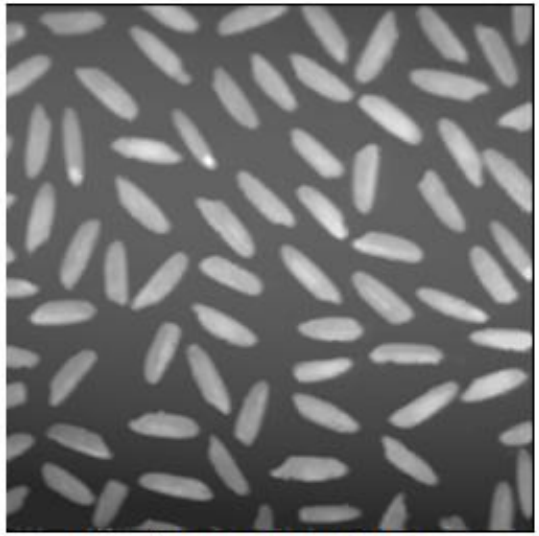
```
A = imread('board.tif');
A = rgb2gray(A(1:300,1:300,:));
figure,imshow(A),title('ORIGINAL IMAGE');
B=zeros(size(A));
modifyA=padarray(A,[1 1]);
     x=[1:3]';
     y=[1:3]';
for i= 1:size(modifyA,1)-2
   for j=1:size(modifyA,2)-2
     window=reshape(modifyA(i+x-1,j+y-1),[],1);
     B(i,j)=max(window);
   end
end
B=uint8(B);
figure,imshow(B),title('IMAGE AFTER MAX FILTERING');
```

ORTA-DEĞER (MEDİAN) OPERATÖRÜ (NONLİNEER FİLTRELEMEDİR)

- min ve max operatörlerinin fonksiyonlarını birleştiren ve onların yukarıda bahsedilen dezavantajlarına sahip olmayan bir operatöre ihtiyaç vardır. Şekil bu çözümü göstermektedir.
- Orta-değer operatöründeki fikir, maske içerisindeki gritonları değerlerine göre sıralamaktır. Bu sıralamada ortada bulunan değer sonuç resmin ilgilenilen pikseli için kullanılır. Bu strateji yüksek ve alçak gritonların piklerini, griton bölgeleri ayıran griton basamaklarını yassılaştırmadan yok eder. Orta-değer operatörünün dezavantajı; komşu piksel gritonlarının sıralamasından dolayı hesaplama zamanın yüksek olmasıdır.
- Orta-değer operatörünün resim üzerindeki etkisini göstermektedir. Resimden de görüleceği üzere, gürültülü resim tamamen onarılmıştır.



Kaynak resim



Orta-değer operatörü uygulanmış sonuç resim

K-ENYAKIN-KOMŞU (NONLİNEER FİLTRELEMEDİR:)

Diğer bir kenar korumalı düzgünleştirme metodu "k-enyakın-komşu" metodudur. Bu, maskenin bütün pikselleri üzerinde kullanılmayan bir normal ortalama operatörüdür (kutu filtre).

Sadece gritonları ilgilenilen pikselin gritonuna en yakın k tane piksel üzerine uygulanır. Şekilde 3x3 enyakın-komşu operatörünün k = 3 ile (ilgilenilen piksel dahil) kaynak resme uygulanması ile elde edilen sonucu göstermektedir. Ortalamanın sadece 3 griton kullanılarak hesaplanmasından dolayı, düzgünleştirme etkisi "orta-değer" operatöründen azdır.

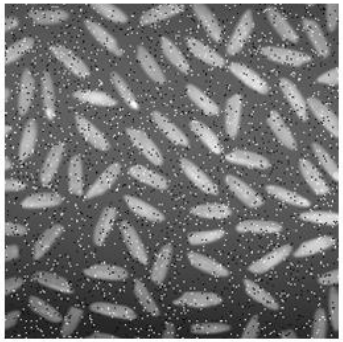
Genellikle k, maskedeki piksel sayısının yarısından fazla olmalıdır. Şekil, k-enyakın-komşu operatörünün resim üzerindeki uygulamasını göstermektedir.

orijinal resim

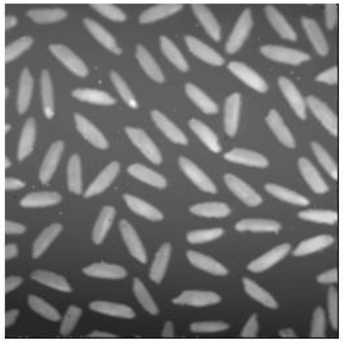
1	1	1	1	10	10	10	10
1	1	6	1	8	10	2	10
1	3	1	1	9	10	7	10
1	1	1	2	8	9	10	10
1	1	1	1	10	10	10	10
1	4	1	2	9	10	2	10
1	2	1	8	10	10	10	10
1	1	1	1	10	10	10	10

0	0	0	0	0	0	0	0
0	1	3	1	9	10	6	0
0	2	1	1	9	10	9	0
0	1	1	1	9	9	10	0
0	1	1	1	10	10	10	0
0	2	1	1	10	10	7	0
0	1	1	9	10	10	10	0
0	0	0	0	0	0	0	0

3x3 en yakın komşu operatörü uygulanmış sonuç resim (k=3)



Kaynak resim



k-enyakın-komşu operatörü uygulanmış sonuç resim

NONLİNEER UZAYSAL FİLTRE FONKSİYONLARI

- Matlab image processing toolbox'ta(IMT) nonlinear filtreleme için iki önemli fonksiyon vardır:

  nlfilter

  colfilt

Her ikiside doğrudan 2 boyutta çalışır.

colfilt; datayı sütun formunda organize eder. Daha fazla hafızaya ihtiyaç gösterir. nfilter fonksiyonundan daha hızlı çalışır.

Birçok görüntü işleme uygulamalarında hız önemli bir faktördür. Nonlineer uzaysal filtreleme işlemlerinde daha çok colfilt kullanılır.

```
g=colfilt(f,[m n],'sliding',@fun,parameters)
```

m ve n filtre boyutudur.

'sliding', f giriş görüntüsünde pikselden piksele mxn'lik filtrenin kaydırma işlemini belirtir.

@fun; references a function, which we denote arbitrarily as fun,

parameters indicates parameters (separated by commas) that may be required by function fun.

The symbol @ is called a *function handle, a MATLAB data type* that contains information used in referencing a function.

```
fp=padarray(f,[r c],method,direction)
```

- Lineer filtrelerde söylendiği gibi, görüntünün kenarlarınada filtrenin uygulanabilmesi için görüntü datasına genellikle 0'lar eklenmeliydi.
- Colfilt kullanılacaksa, filtrelemeden önce muhakkak giriş datasına yeterli miktarda ekleme yapılmalıdır. Bunun için padarray fonksiyonu kullanılır.

  Burada f giriş görüntüsü, fp paddlenmiş(eklenmiş) görüntü,[r c], f'ye eklenecek pad'in boyutunu verir.

  method ve direction yeni tablodaki yerleşimi bildirir.

Options	Description
Method	
'symmetric'	The size of the image is extended by mirror-reflecting it across its border.
'replicate'	The size of the image is extended by replicating the values in its outer border.
'circular'	The size of the image is extended by treating the image as one period of a 2-D periodic function.
Direction	
'pre'	Pad before the first element of each dimension.
'post'	Pad after the last element of each dimension.
'both'	Pad before the first element and after the last element of each dimension. This is the default.

```
>> A=[1 2 3; 4 5 6]
A =
     1     2     3
     4     5     6
>> B=padarray(A,[2 3],0,'both')
B =
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     1     2     3     0     0     0
     0     0     0     4     5     6     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
```

```
>> B=padarray(A,[2 3],0,'post')
B =
     1     2     3     0     0     0
     4     5     6     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
>> B=padarray(A,[2 3],0,'pre')
B =
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     1     2     3
     0     0     0     4     5     6
```

```
A = [ 1 2; 3 4];
B = [ 5 6; 7 8];
C = cat(3,A,B)
C(:,:,1) =
     1     2
     3     4
C(:,:,2) =
     5     6
     7     8
D = padarray(C,[3 3])
```

D(:,:,1) ==

D(:,:,1) ==

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	2	0	0	0
0	0	0	3	4	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

D(:,:,2) ===

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	5	6	0	0	0
0	0	0	7	8	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

```
a = [ 1 2 3 4 ];
b = padarray(a,[0
3],'symmetric','pre')
b =
```

3	2	1	1	2	3	4

```
>>A = [1 2; 3 4];
>>B = padarray(A,[3
2],'replicate','post')
B =
```

1	2	2	2
3	4	4	4
3	4	4	4
3	4	4	4
3	4	4	4

ÖNEMLİ NOT: FREKANS BİLEŞENİ KAVRAMI

- Bir görüntüdeki filtrenin etkisini standart bir formda ifade edebilmek için en önemli kriterlerden birisi görüntünün frekans bileşenleridir.
- Kabaca bir görüntünün frekans bileşenleri; mesafeye göre gri seviye değişiminin miktarı olarak ifade edilir.
- Görüntünün Yüksek frekanslı bileşenleri; küçük mesafelerde piksellerin gri değerlerin büyük miktarda değişikliklerini karakterize eder. Yüksek frekans bileşenlerine örnek olarak; resmin kenarları (en büyük gri seviye değişimleri kenarlarda olur) ve gürültüler verilebilir.
- Görüntününün Düşük frekanslı bileşenleri: Resimdeki piksellerin gri değerlerinin mesafeye göre pek az değiştiği görüntü parçaları ile karakterize edilir. Bunlara örnek arka planlar (gri seviyeleri çok az değişen yüzeyler), cilt dokuları verilebilir.
- Bu tanımlamalar göre filtreleri;
- Yüksek geçiren Filtreler: Yüksek frekanslı bileşenleri geçirir. Düşük frekanslı bileşenleri yok eder. Örnek kenar çıkarma işlemleri.
- Alçak geçiren Filtreler: Görüntüdeki Alçak frekans bileşenlerini geçirir veya kuvvetlendirir. Yüksek frekans bileşenlerin yok eder. Örneğin 3x3Lük average filtresi bir alçak geçiren filtredir. Çünkü resmin kısa mesafede değişen (Özellikle kenarlar - Gürültüler) piksellerini yok eder. Belirli mesafede az değişen piksellerini muhafaza eder. Dolayısıyla resim bulanıklaşır (Blurring).

FSPECİAL FONKSİYONU

- Lineer uzaysal filtrelemenin diğer bir yolu da özel lineer filtrelerin yaratılabilmesidir. Buna göre, oluşturulacak lineer filitre maskesi için aşağıdaki deyim kullanılır.

```
>> F = fspecial ('type', parameters)
```

Burada 'type', özel filtre tipini belirtir. Parameters ise filtreyi tanımlayan değerlerdir. Bu şekilde kullanabileceğimiz hazır özel filtreler vardır. Bunlar tabloda görülmektedir.

ÇOK KULLANILAN BAZI ÖZEL LİNEER FİLTRELERİN FSPECİAL İLE TANIMI

Average: Ağırlıklı ortalama filtredir. Default boyutu 3x3 boyutlu elemanlar 1 olan bir matristir. Eğer r,c yerine tek bir sayı yazılırsa kare matris filtre olur.

Disk: Bir dairesel ortalama filtredir. r dairenin yarıçapıdır. R'nin default değeri 5'tir.

Gaussian: Alçak geçiren bir filtredir. Standart sapması pozitif olan ve boyutu rxc olan bir filtredir. Rxc eğer tek bir değer ise kare matris filtredir.

Laplacian ve log, 2.türev işlemine dayanan yüksek filtre tipleridir.

Type	Syntax and Parameters
`'average'`	`fspecial('average',[r c])`. A rectangular averaging filter of size `r×c`. The default is 3×3. A single number instead of `[r c]` specifies a square filter.
`'disk'`	`fspecial('disk',r)`. A circular averaging filter (within a square of size 2r+1) with radius `r`. The default radius is 5.
`'gaussian'`	`fspecial('gaussian',[r c],sig)`. A Gaussian lowpass filter of size `r×c` and standard deviation `sig` (positive). The defaults are 3×3 and 0.5. A single number instead of `[r c]` specifies a square filter.
`'laplacian'`	`fspecial('laplacian',alpha)`. A 3×3 Laplacian filter whose shape is specified by `alpha`, a number in the range $[0, 1]$. The default value for `alpha` is 0.5.
`'log'`	`fspecial('log',[r c],sig)`. Laplacian of a Gaussian (LoG) filter of size $rtimesc$ and standard deviation `sig` (positive). The defaults are 5×5 and 0.5. A single number instead of `[r c]` specifies a square filter.

ÇOK KULLANILAN BAZI ÖZEL LİNEER FİLTRELERİN FSPECİAL İLE TANIMI (2)

Previt ve sobel Filtreleri yüksek geçiren basit tip filtrelerdir. Görüntüde kenar çıkarma, kenar iyileştirme işlemlerinde kullanılabilirler.

Type	Syntax and Parameters
`'motion'`	`fspecial('motion',len,theta)`. Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of `len` pixels. The direction of motion is `theta`, mesaured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.
`'prewitt'`	`fspecial('prewitt')`. Outputs a 3×3 Prewitt mask, `wv`, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: `wh=wv'`.
`'sobel'`	`fspecial('sobel')`. Outputs a 3×3 Sobel mask, `sv`, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: `sh=sv'`.
`'unsharp'`	`fspecial('unsharp',alpha)`. Outputs a 3×3 unsharp filter. Parameter `alpha` controls the shape; it must be greater than or equal to 0 and less than or equal to 1.0; the default is 0.2.

```
>> fspecial('average',[3 3])

ans =

    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111

>> a=fspecial('disk',3)

a =

         0    0.0003    0.0110    0.0172    0.0110    0.0003         0
    0.0003    0.0245    0.0354    0.0354    0.0354    0.0245    0.0003
    0.0110    0.0354    0.0354    0.0354    0.0354    0.0354    0.0110
    0.0172    0.0354    0.0354    0.0354    0.0354    0.0354    0.0172
    0.0110    0.0354    0.0354    0.0354    0.0354    0.0354    0.0110
    0.0003    0.0245    0.0354    0.0354    0.0354    0.0245    0.0003
         0    0.0003    0.0110    0.0172    0.0110    0.0003         0

>> fspecial('gaussian',[3,3],0.5)

ans =

    0.0113    0.0838    0.0113
    0.0838    0.6193    0.0838
    0.0113    0.0838    0.0113
```

```
>> a=fspecial('prewit')

a =

     1     1     1
     0     0     0
    -1    -1    -1

>> a=fspecial('sobel')

a =

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

>> a=fspecial('motion',9,90)

a =

    0.1111
    0.1111
    0.1111
    0.1111
    0.1111
    0.1111
    0.1111
    0.1111
    0.1111
```

ÇOK KULLANILAN ÖZEL NONLİNEER UZAYSAL FİLTRELER

```
g=ordfilt2(f,order,domain
```

Matlabda; özel Nonlineer uzaysal filtre işlemi genel olarak ordfilt2 fonksiyonuyla gerçekleştirilir. Bu filtre dereceli (sıralı)- istatistiksel filtreleme (rank –order- sayı sıralı-Filtre) işlemidir.

Bu fonksiyon 3 parametreye ihtiyaç gösterir.

F: filtrelenecek görüntü matrisi,

order; rakamsal bir değerdir. Maske içindeki en küçük veya en büyük n.değere göre işlem yapmayı ifade eder.

Domain; maske boyutunun ve elemanlarının tanımı'dır.

En çok bilinenleri, min filtre, Maks.Filtre ve Median Filtredir.

g=ordfilt2(f, 1, ones(m,n))

İşlemi, mxn boyutlu ve elemanları 1 olan bir maske kullanılarak, f görüntüsünü min filtreye tabi tutmaktır. g'nin ilgili elemanını değeri maskenin kapsadığı komşu elemanlar içerisinde en küçük değerde olanıdır(order=1 olduğu için).

MİN.FİLTRELEME ÖRNEĞİ

```
>> B=[2 14 26 37 99;34 90 67 56 100; 100 200 56 255 45;
      10 100 56 178 35; 99 29 0 150 200]
B =

     2    14    26    37    99
    34    90    67    56   100
   100   200    56   255    45
    10   100    56   178    35
    99    29     0   150   200

>> d=uint8(B)

d =

     2    14    26    37    99
    34    90    67    56   100
   100   200    56   255    45
    10   100    56   178    35
    99    29     0   150   200

>> c=ordfilt2(d,1,ones(3,3))

c =

     0     0     0     0     0
     0     2    14    26     0
     0    10    56    35     0
     0     0     0     0     0
     0     0     0     0     0
```

```
>> c=ordfilt2(d,2,ones(3,3))

c =

     0     0     0     0     0
     0    14    26    37     0
     0    34    56    45     0
     0    10    29    35     0
     0     0     0     0     0
```

```
>> c=ordfilt2(d,4,ones(3,3))

c =

     0     2    14    26     0
     2    34    56    56    37
    10    56    67    56    35
    10    56    56    56    35
     0     0     0     0     0
```

MAKS.FİLTRELEME ÖRNEĞİ
G=ORDFİLT2(F,M*N,DOMAİN)

Maks.filtre maskenin kapsadığı elemanlar içindeki en büyük değerli elemanı yeni elemanla değiştirmek olduğuna göre, min filtrede (m*n) değerini order kısmına yazmak yeterlidir.

```
>> B=[2 14 26 37 99;34 90 67 56 100; 100 200 56 255 45;
      10 100 56 178 35; 99 29 0 150 200]
B =

     2    14    26    37    99
    34    90    67    56   100
   100   200    56   255    45
    10   100    56   178    35
    99    29     0   150   200

>> d=uint8(B)

d =

     2    14    26    37    99
    34    90    67    56   100
   100   200    56   255    45
    10   100    56   178    35
    99    29     0   150   200

>> c=ordfilt2(d,9,ones(3,3))

c =

    90    90    90   100   100
   200   200   255   255   255
   200   200   255   255   255
   200   200   255   255   255
   100   100   178   200   200
```

MEDİAN FİLTRELEMEYE ÖRNEK
G=ORDFİLTER2(F, MEDİAN (1:M*N), ONES(M,N))

Median Filtre maske altındaki piksel gri seviye değerlerini azdan çoğa doğru bu değerlerin ortadakini almaktı. Yukarıdaki fonksiyonda order yerine (1:m*n) değerleri arasında %50 büyüklükteki değerin alınmasıdır. Median filtreler görüntüdeki salt-pepper(tuz-Biber) gürültülerini filtreler.

```
>> B=[2 14 26 37 99;34 90 67 56 100;100 200 56 255 45;
      10 100 56 178 35; 99 29 0 150 200]
B =

     2    14    26    37    99
    34    90    67    56   100
   100   200    56   255    45
    10   100    56   178    35
    99    29     0   150   200

>> c=ordfilt2(d,median(1:3*3),ones(3,3))

c =

     0    14    26    37     0
    14    56    56    56    45
    34    67    90    56    45
    29    56   100    56    45
     0    10    29    35     0
```

MEDİAN FİLTREYE ÖRNEK

```
>> a=imread('cameraman.tif');
>> imshow(a)
>> b=imnoise(a,'salt & pepper',0.2);
>> imshow(b)
>> c=ordfilt2(b,median(1:3*3),ones(3,3));
>> imshow(c)
>> d=ordfilt2(b,median(1:9*9),ones(9,9));
>> imshow(d)
```

- Burada mxn uzunlukta bir komşuluk ilişkisi tanımlanır ve onun piksel değerlerinin ortancası ile hesaplar yapılır.
- Padopt üç olası sınır padding seçeneklerden birini belirtir:
- Sıfır (varsayılan-default): kenarlara 0 dolgusu yapılır.
- Simetrik: kenarlar ayna-yansımalı simetriklik özelliği ile doldurulur.
- 'endeksleme' ise, veri double ise kenarlar 1 ile değil ise 0'la doldurulur.

Örnek default fonksiyon

g = medfilt2 (f)

3x3 maske ile medyan hesaplar. Kenarlar 0 ile doldurulur.

MEDİAN FİLTRE İÇİN BAŞKA BİR FONKSİYON

G=MEDFİLT2(F, [M N], PADOPT)

ÖRNEK:medfilt2

```
 I = imread('eight.tif');
J = imnoise(I,'salt & pepper',0.02);
K = medfilt2(J);
imshow(J), figure, imshow(K)
```

- Medfilt2 fonksiyonunun medyan filtreleme için yaptığı işi gerçekleştiren bir MATLAB fonksiyonu yazınız?
- Derste gösterilen filtrelerin MATLAB uygulamalarını gerçekleştirerek bir rapor hazırlayınız .

WORD VE MATLAB DOSYASINI RAR DOSYASI İÇİNE ATIP YÜKLEYİN.

ÖDEV:
TESLİM TARİHİ: 29.04.2025