



GÖRÜNTÜ İŞLEME DERS-10 YARDIMCI NOTLARI -2019

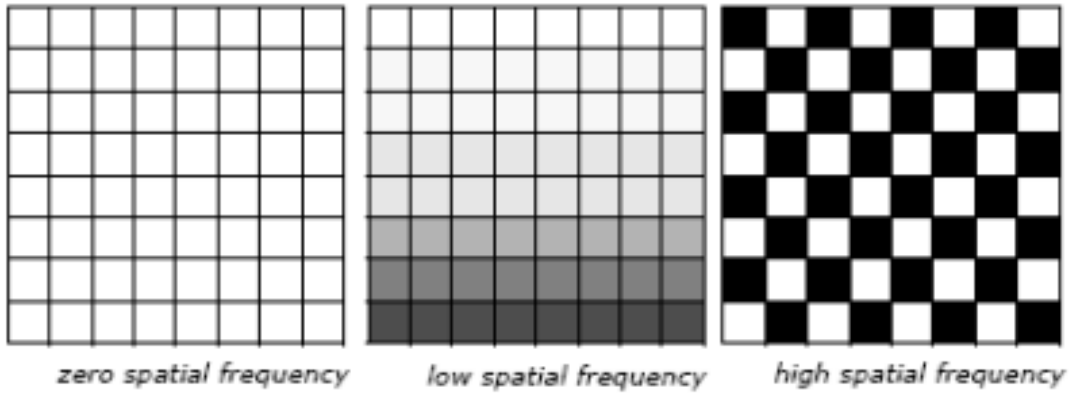
GÖRÜNTÜ FİLTRELEME

Filtreler görüntü zenginleştirme amacı ile de uygulanan, adından da anlaşılacağı gibi görüntüde belirli ayrıntıların ayıklanması ya da daha belirgin hale getirilmesi vb. gibi operasyonları gerçekleştiren operatörlerdir.

Farklı amaçlar için farklı filtreleme operatörleri vardır. Bunlara:

- Gürültü Yok etme
- Kenar keskinleştirme
- Kenar yakalama
- Görüntü yumuşatma ve bunun gibi daha birçok amaçla kullanılan filtreler örnek verilebilir.

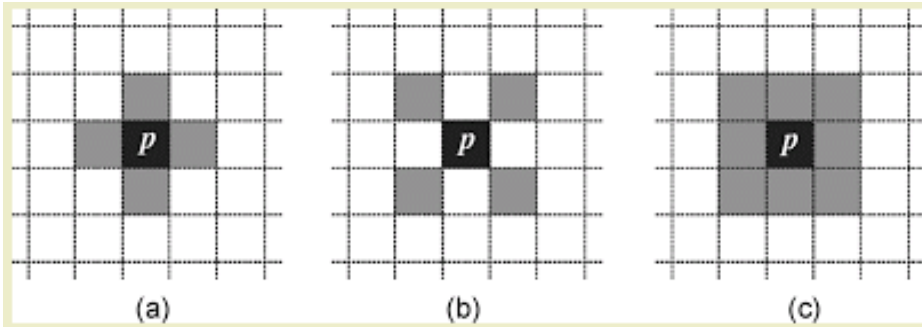
Uzaysal frekanslar:



PİKSEL KOMŞULUKLARI:

Bir piksel 1. Derece, 2. Derece... n. Derece komşu pikselleri ile birlikte, tek başına ifade ettiği değerden daha fazla değer ifade eder. Yani komşu pikselleri ile birlikte bir pikselden çok daha fazla akıllı bilgi çıkartabiliriz.

Aşağıda bir pikselin (p) çeşitli derece komşuları gösterilmiştir.



a) 4' lü komşuluk b) diagonal komşuluk c) 8' li komşuluk



KORELASYON VE KONVOLÜSYON KAVRAMLARI :

- Konvolüsyon ve korelasyonda 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.

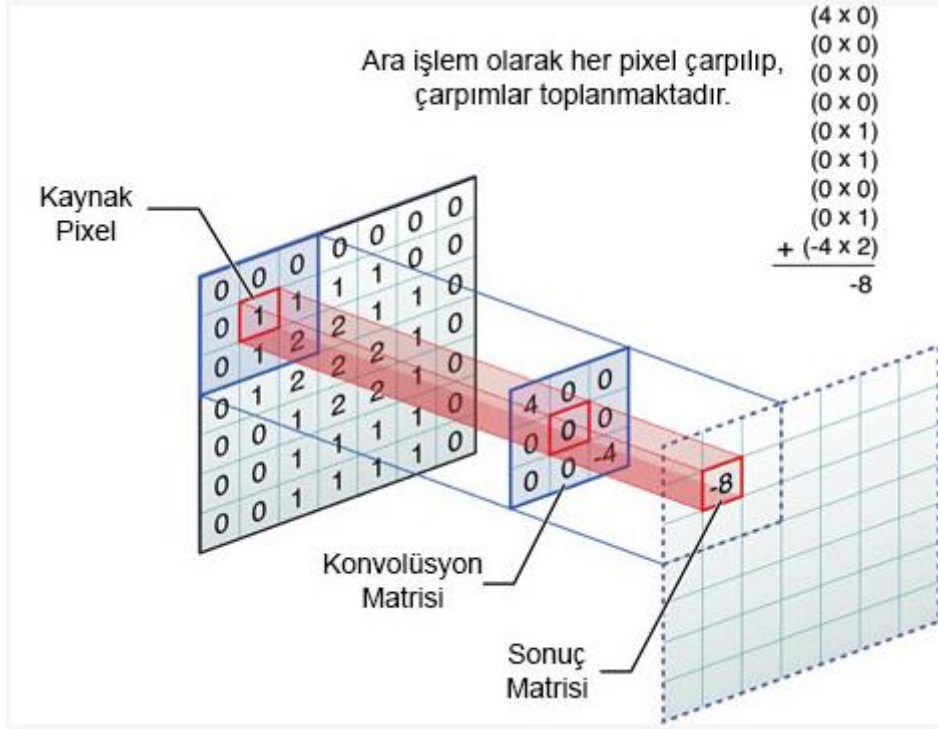
Bir boyutlu korelasyon:

Resimde f dizisinin, w dizisi ile korelasyon işlemi görülmektedir. İki fonksiyonun korelasyonunu başarmak için;

- w dizisinin en sağdaki noktasının, f dizisinin en soldaki noktasına gelecek şekilde konumlandırılarak 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. (Zero padding)
- Böylelikle f dizisinin her elemanınının w dizisi ile ilişkiye girmesi garanti edilmiş olur.

(BMÜ-357 Sayısal Görüntü İşleme Yrd. Doç. Dr. İlhan AYDIN)

1	<p>(a) \swarrow Origin f w 0 0 0 1 0 0 0 0 1 2 3 2 0</p>
2	<p>(b) \downarrow 0 0 0 1 0 0 0 0 1 2 3 2 0 \nwarrow Starting position alignment</p>
3	<p>(c) Zero padding 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 0</p>
4	<p>(d) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 0 \nwarrow Position after one shift</p>



Sonuç olarak korelasyon ve konvolüsyonun matematiksel eşitliği:

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

Burada: $w(x, y)$ filtre matrisi, m ve n filtre matrisi boyutlarıdır.

$$a = (m-1) / 2$$

$$b = (n-1) / 2$$

$w(x, y)$ filtre matrisi.

Bir boyutlu konvolüsyon işlemi:

$$u = [1 \ 1 \ 1];$$

$$v = [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1];$$

$$w = \text{conv}(u, v)$$

$$w = \text{conv}(u, v, 'same')$$



İki boyutlu konvolüsyon işlemi

```
x=zeros (7)
```

```
x =
```

```
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 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

```
>> x(4,4)=1
```

```
x =
```

```
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 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

```
>> y=[1 2 3;4 5 6;7 8 9]
```

```
y =
```

```
1 2 3
4 5 6
7 8 9
```

```
>> conv2(x,y,'same')
```

```
ans =
```

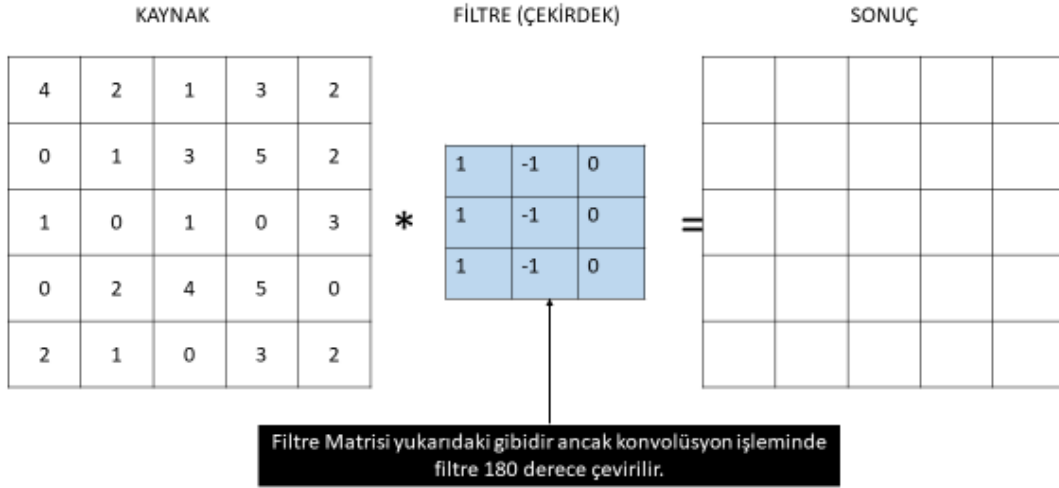
```
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 1 2 3 0 0
0 0 4 5 6 0 0
0 0 7 8 9 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ

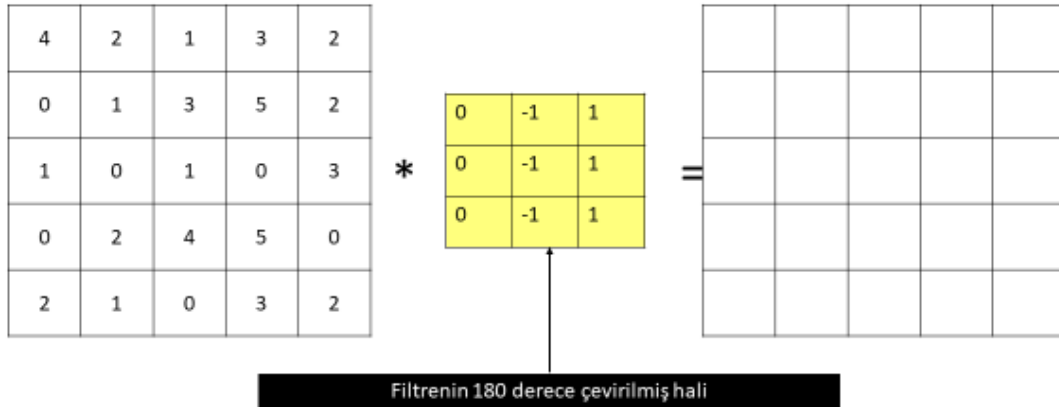


Konvolüsyon işlemi ayrıntılı örnek gösterimi:
1. adım:



2. adım:

KONVOLÜSYON İŞLEMİ





AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



3. adım:

ZERO PADDING (SIFIR İLE DOLDURMA)

0	0	0	0	0	0	0
0	4	2	1	3	2	0
0	0	1	3	5	2	0
0	1	0	1	0	3	0
0	0	2	4	5	0	0
0	2	1	0	3	2	0
0	0	0	0	0	0	0

*

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

=

4. adım:

0	-1	1	0	0	0	0
0	0	0	0	0	0	0
0	-1	1	1	3	2	0
0	0	1	3	5	2	0
0	1	0	1	0	3	0
0	0	2	4	5	0	0
0	2	1	0	3	2	0
0	0	0	0	0	0	0

*

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

=

-1				

$$0x0 + (-1)x0 + 1x0 + 0x0 + (-1)x4 + 1x2 + 0x0 + 0x(-1) + 1x1 = -1$$



5. adım:

0	0	-1	1	0	0	0
0	4	2	1	3	2	0
0	0	1	3	5	2	0
0	1	0	1	0	3	0
0	0	2	4	5	0	0
0	2	1	0	3	2	0
0	0	0	0	0	0	0

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

-1	1			

$$0x0 + (-1)x0 + 1x0 + 4x0 + 2x(-1) + 1x1 + 0x0 + (-1)x(-1) + 1x3 = 1$$

6. adım:

0	0	0	0	0	0	0
0	4	2	1	3	2	0
0	0	1	3	5	2	0
0	1	0	1	0	3	0
0	0	2	4	5	0	0
0	2	1	0	3	2	0
0	0	0	0	0	0	0

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

-1	1	4		

$$0x0 + (-1)x0 + 1x0 + 0x2 + (-1)x(-1) + 1x3 + 0x1 + (-1)x3 + 1x5 = 4$$



7. adım:

0	0	0	0	-1	1	0	0
0	4	2	1	3	2	0	0
0	0	1	3	5	2	0	0
0	1	0	1	0	3	0	0
0	0	2	4	5	0	0	0
0	2	1	0	3	2	0	0
0	0	0	0	0	0	0	0

*

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

=

-1	1	4	-4	

$$0 \times 0 + (-1) \times 0 + 1 \times 0 + 0 \times 1 + (-1) \times (3) + 1 \times 2 + 0 \times 3 + (-1) \times 5 + 1 \times 2 = -4$$

8. adım:

0	0	0	0	0	-1	1	0
0	4	2	1	3	2	0	0
0	0	1	3	5	2	0	0
0	1	0	1	0	3	0	0
0	0	2	4	5	0	0	0
0	2	1	0	3	2	0	0
0	0	0	0	0	0	0	0

*

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

=

-1	1	4	-4	-4

$$0 \times 0 + (-1) \times 0 + 1 \times 0 + 0 \times 3 + (-1) \times 2 + 1 \times 0 + 0 \times 5 + (-1) \times 2 + 1 \times 0 = -4$$



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



9. adım:

0	0	0	0	0	0	0	0
0	-1	1		1	3	2	0
0	-1	1		3	5	2	0
0	-1	1		1	0	3	0
0	0	2	4	5	0	0	0
0	2	1	0	3	2	0	0
0	0	0	0	0	0	0	0

*

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

=

-1	1	4	-4	-4
-2				

$$0x0 + (-1)x4 + 1x2 + 0x0 + (-1)x0 + 1x1 + 0x0 + (-1)x1 + 1x0 = -2$$

10. adım:

0	0	0	0	0	0	0	0
0	0	-1	1		3	2	0
0	0	-1	1		5	2	0
0	0	-1	1		0	3	0
0	0	2	4	5	0	0	0
0	2	1	0	3	2	0	0
0	0	0	0	0	0	0	0

*

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

=

-1	1	4	-4	-4
-2	2			

$$0x4 + (-1)x2 + 1x1 + 0x0 + (-1)x1 + 1x3 + 0x1 + (-1)x0 + 1x1 = 2$$



11. adım:

0	0	0	0	0	0	0
0	4	0	-1	1	2	0
0	0	0	-1	1	2	0
0	1	0	-1	1	3	0
0	0	2	4	5	0	0
0	2	1	0	3	2	0
0	0	0	0	0	0	0

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

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

$$0x2 + (-1)x1 + 1x3 + 0x1 + (-1)x3 + 1x5 + 0x0 + (-1)x1 + 1x0 = 3$$

. . . son adım:

0	0	0	0	0	0	0
0	4	2	1	3	2	0
0	0	1	3	5	2	0
0	1	0	1	0	3	0
0	0	2	4	0	-1	1
0	2	1	0	0	-1	1
0	0	0	0	0	-1	1

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

-1	1	4	-4	-4
-2	2	3	-1	-7
2	5	2	-5	-5
0	2	3	-3	-5
1	1	4	-6	-2

$$0x5 + (-1)x0 + 1x0 + 0x3 + (-1)x2 + 1x0 + 0x0 + (-1)x0 + 1x0 = -2$$

Konvolüsyon işlemi tamamlandı. Sonuçta kaynak görüntü matrisi farklı bir matris haline dönüşmüş oldu.



KONVOLÜSYON ÖRNEĞİ:

Bir adet 10x10 a görüntüsü hazırlayalım.

```
0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 1 1 1 1 1
0 1 0 1 0 0 0 0 0 0
0 1 0 0 1 0 0 0 0 0
0 1 0 0 0 1 0 0 0 0
0 1 0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 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 0 0 0 0 0
```

>>figure,imshow(a);

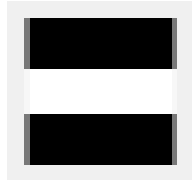


Yukarıdaki resimde yatay, dikey ve diagonal doğru parçaları mevcuttur. Amacımız doğru filtreler seçerek konvolüsyon işlemi yapmak ve resimdeki istediğimiz kısmı (yatay, dikey ve diagonal) parlatmak, istemediğimiz kısımları ise sönmülemektir.

Bunun için filtrelerimizi oluşturalım.

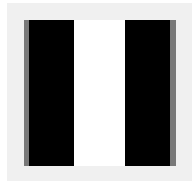
```
yatayFiltre =
```

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



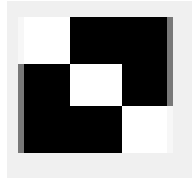
```
dikeyFiltre =
```

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



```
diagFiltre =
```

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



Yatay filtre kullanarak a görüntü matrisini konvolüsyon yaptığımızda görüntüdeki yatay kısmın parlamasını ve diğer kısımların sönmüşmesini bekliyoruz.

>>yatay=conv2(a,yatayFiltre,'same');



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



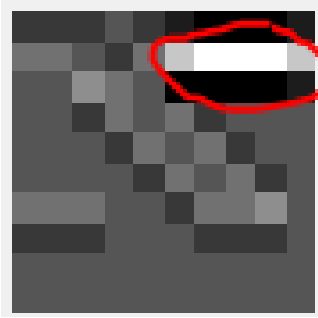
yatay =

```
-1 -1 -1 0 -1 -2 -3 -3 -3 -2  
1 1 0 -1 1 4 6 6 6 4  
0 0 2 1 0 -3 -3 -3 -3 -2  
0 0 -1 1 0 1 -1 0 0 0  
0 0 0 -1 1 0 1 -1 0 0  
0 0 0 0 -1 1 0 1 -1 0  
1 1 1 0 0 -1 1 1 2 0  
-1 -1 -1 0 0 0 -1 -1 -1 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0
```

Sonuç matrisini görüntüleyebilmek için ve negatif değerlerden kurtulabilmek için sonuç matrisini sıfır ile bir aralığına ölçekleyelim.

```
>>yatay=mat2gray(yatay);
```

```
>>imshow(yatay);
```

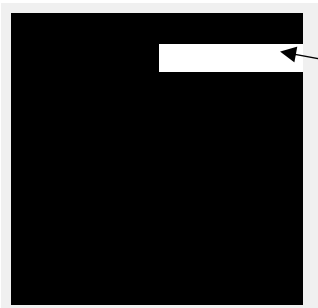


Parlayan kısım kaynak resimdeki yatay kısımdır.

Yukarıdaki resmi siyah-beyaz yani ikilik görüntüye çevirelim.

```
>> yatayBW=im2bw(yatay,0.6);
```

```
>> figure, imshow(yatayBW);
```



Yatay kısım arka plandan tamamen ayrıldı.



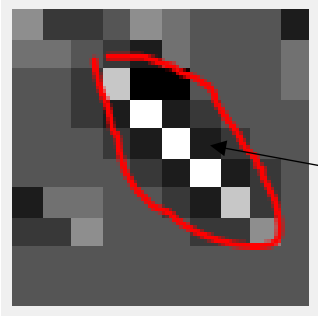
AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



Aynı şekilde diagonal kısmı bulalım:

```
>>diag=conv2(a,diagFiltre,'same');
```

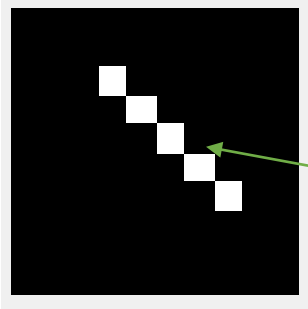
```
>> diag=mat2gray(diag); imshow(diag);
```



Parlayan kısım kaynak resimdeki
diagonal kısımdır.

```
>> diagBW=im2bw(diag,0.6);
```

```
>> figure, imshow(diagBW)
```



Diagonal kısım arka plandan
tamamen ayrıldı.