



GÖRÜNTÜ İŞLEME DERS-6 YARDIMCI NOTLARI -2018

İMGE MANTIKSAL FONKSİYONLARI

İmge matrisleri arasında aritmetik işlemler yapılabildiği gibi

Mantıksal işlemler de yapılabilmektedir. Bu mantıksal işlemler piksel parlaklık değerleri arasında bit – bit karşılığında yapılır. Örneğin parlaklık değeri 255 olan bir pikselin 8 bit karşılığı “1 1 1 1 1 1 1 1” dir. Diğer pikselin parlaklık değeri 3 ise bunun da 8 bit karşılığı “0 0 0 0 0 1 1” dir. İkisini AND işlemine tabi tutarsak sonucu 3’dür. OR işlemine tabi tutarsak sonucu 255’dir.

Bu sonuçlar ışığında mantıksal işlemler genellikle bir görüntünün maskelenmesi amacıyla kullanılırlar.

Ancak mantıksal işlemlerde uyulması gereken bir takım kurallar vardır. Öncelikle iki görüntünün boyutları birbirine eşit olmalıdır. Ayrıca mantıksal işlem yapılacak piksel parlaklık değerlerinin 8 bit integer yani uint8 olması gerekir.

AND, NAND

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NAND

Örnek: bir görüntüde küçük bir alanın kesilip çıkartılması.
logical1.m

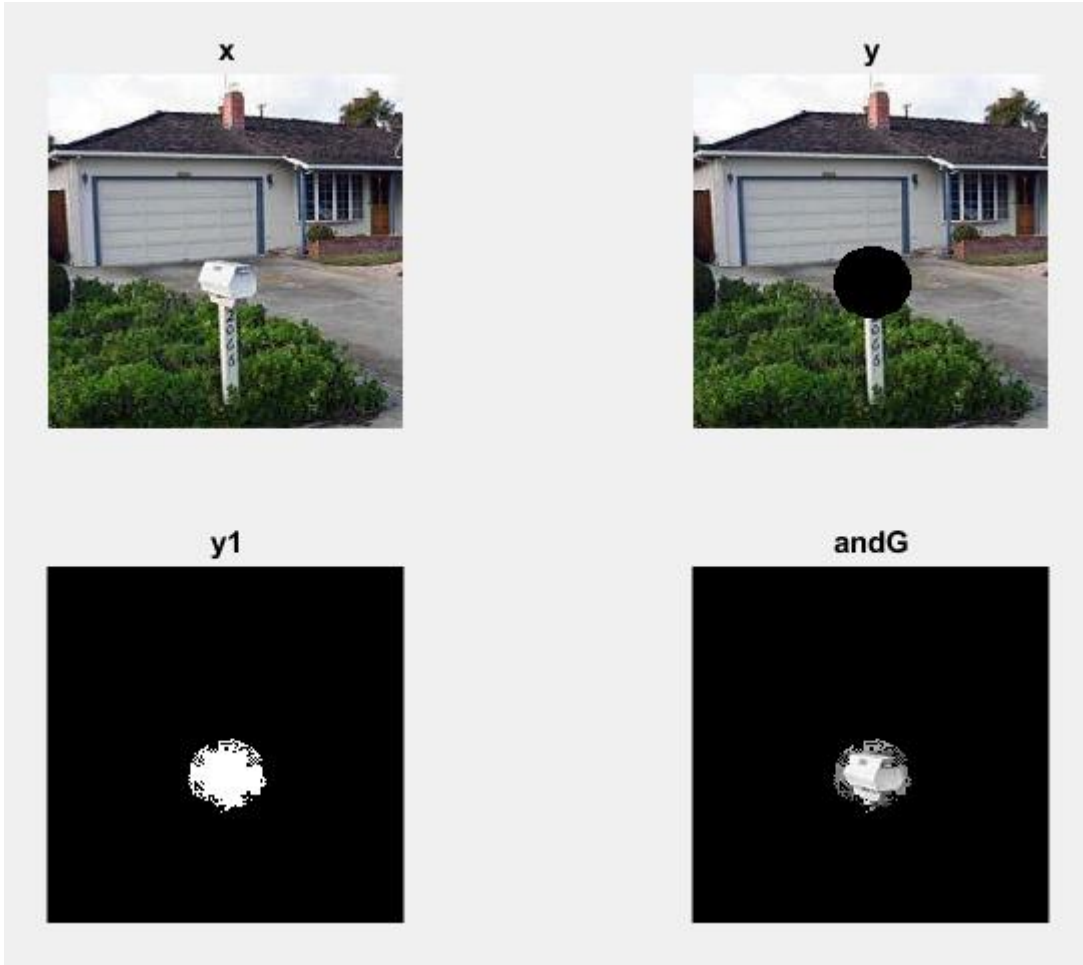
```
1   clc;clear;close all;
2   x=imread('headquarters-2.jpg');
3   y=imread('headquarters-2AND.jpg');
4   subplot(2,2,1)
5   imshow(x)
6   title('x')
7   subplot(2,2,2)
8   imshow(y)
9   title('y')
10  x=rgb2gray(x);
11  y=rgb2gray(y);
12  y1=y<1;
```



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



```
13 subplot(2,2,3)
14 imshow(y1)
15 title('y1')
16 y2=im2uint8(y1);
17 andG=bitand(x,y2);
18 subplot(2,2,4)
19 imshow(andG)
20 title('andG')
```



Yukarıdaki örnek programda y1 maske görüntü matrisi ile x görüntüsündeki belirli bir alan kopyalanıp alınabilmektedir.

Aritmetik ve mantıksal fonksiyonlar kullanılarak yapılan işlemler:

```
% lojik operatörler
clc;
clear all;
close all;
A=imread('cameraman.tif');
subplot(5,3,1)
```

```
%Aritmetik operatörler
clc;
clear all;
close all;
A=imread('cameraman.tif');
```



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



```
imshow(A)
title('Image A');
B=imread('rice.png');
subplot(5,3,2)
imshow(B)
title('Image B');
C=zeros(size(A));
for (x= 100: 200)
    for (y=100: 200)
        C(x,y)=255;
    end
end
subplot(5,3,3)
imshow(C)
title('Image C');
C=uint8(C);

lr1=bitand(A,C);
subplot(5,3,4)
imshow(lr1)
title('A and C');
lr2=bitand(B,C);
subplot(5,3,5)
imshow(lr2)
title('B and C');
lr3=bitand(A,B);
subplot(5,3,6)
imshow(lr3)
title('A and B');
lr4=bitor(A,C);
subplot(5,3,7)
imshow(lr4)
title('A or C');
lr5=bitor(B,C);
subplot(5,3,8)
imshow(lr5)
title('B or C');
lr6=bitor(A,B);
subplot(5,3,9)
imshow(lr6)
title('A or B');
lr7=bitxor(A,C);
subplot(5,3,10)
imshow(lr7)
title('A exor C');
lr8=bitxor(B,C);
subplot(5,3,11)
imshow(lr8)
title('B exor C');
lr9=bitxor(A,B);
subplot(5,3,12)
imshow(lr9)
title('A exor B');
lr10= bitcmp(A);
subplot(5,3,13)
imshow(lr10)
```

```
B=imread('rice.png');

C=zeros(size(A));
for (x=100:200)
    for (y=100:200)
        C(x,y)=255;
    end
end
C=uint8(C);

lr13=imadd(A,C);
figure,subplot(4,3,1)
imshow(uint8(lr13))
title('A + C');
lr14=imadd(B,C);
subplot(4,3,2)
imshow(uint8(lr14))
title('B + C');
lr15=imadd(A,B);
subplot(4,3,3)
imshow(uint8(lr15))
title('A + B');
lr16=imsubtract(A,C);
subplot(4,3,4)
imshow(uint8(lr16))
title('A - C');
lr17=imsubtract(B,C);
subplot(4,3,5)
imshow(uint8(lr17))
title('B - C');
lr18=imsubtract(A,B);
subplot(4,3,6)
imshow(uint8(lr18))
title('A - B');
lr19=immultiply(A,C);
subplot(4,3,7)
imshow(uint8(lr19))
title('A * C');
lr20=immultiply(B,C);
subplot(4,3,8)
imshow(uint8(lr20))
title('B * C');
lr21=immultiply(A,B);
subplot(4,3,9)
imshow(uint8(lr21))
title('A * B');
lr22=imdivide(A,C);
subplot(4,3,10)
imshow(uint8(lr22))
title('A/C');
lr23=imdivide(B,C);
subplot(4,3,11)
imshow(uint8(lr23))
title('B/C');
lr24=imdivide(A,B);
subplot(4,3,12)
imshow(uint8(lr24))
```



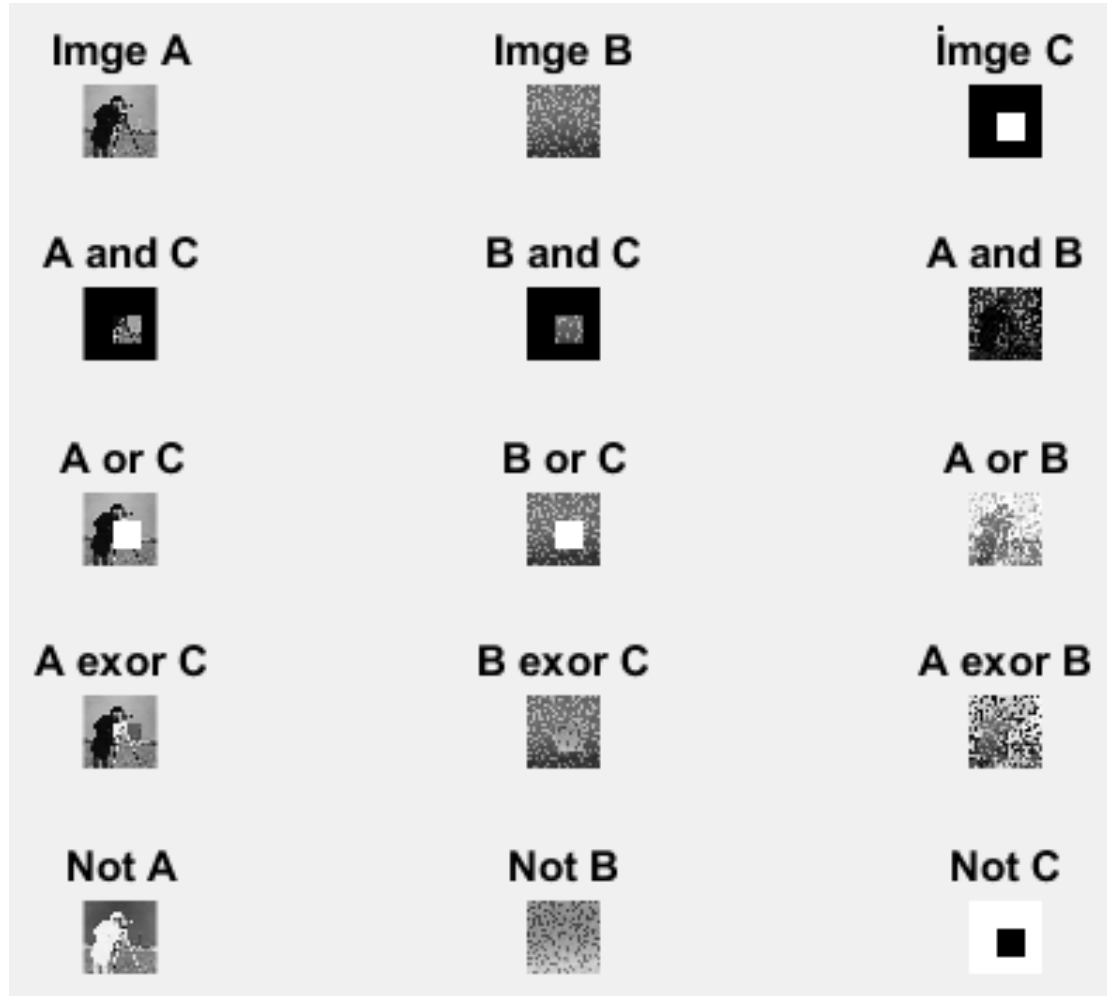
AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



```
title('Not A');  
lr11= bitcmp(B);  
subplot(5,3,14)  
imshow(lr11)  
title('Not B');  
lr12= bitcmp(C);  
subplot(5,3,15)  
imshow(lr12)  
title('Not C');
```

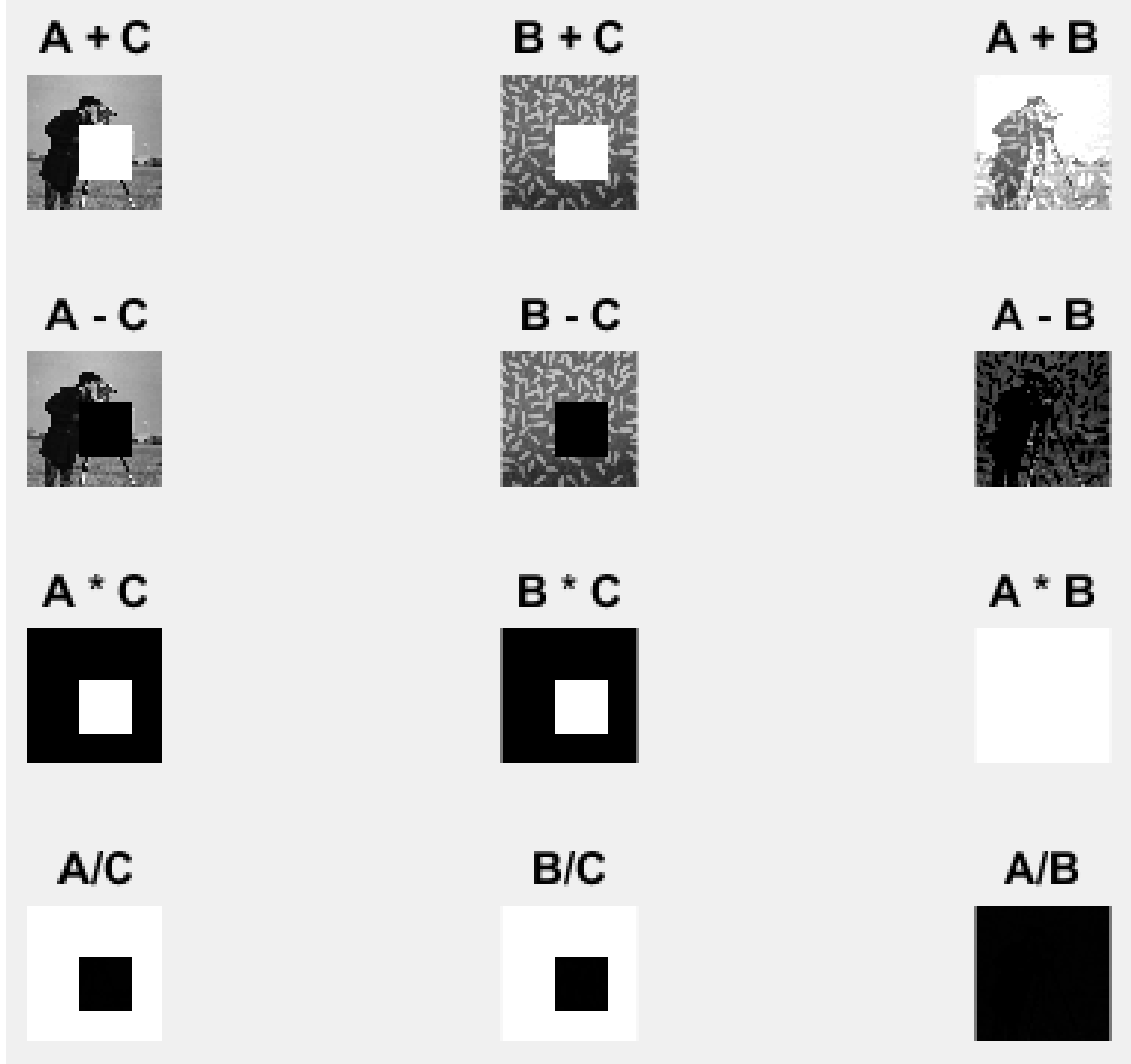
```
title('A/B');
```

Mantıksal işlemlerin sonuçları:





Aritmetik işlemlerin Sonuçları:



Mantıksal ve Aritmetik işlemler arasındaki benzerlikleri araştırabilirsiniz.

İMGE BOYUTLARININ DEĞİŞTİRİLMESİ

Aşağıdaki komutlar ile imge boyutları değiştirilebilir.

imresize() komutu

```
>> A=imread('cameraman.tif');  
>> B=imresize(A,2);  
>> imshow(B)  
>> B=imresize(A,[100,133]); satır 100 sütun 133 'e döndürür.  
>> imshow(B)  
>> B=imresize(A,[133 NaN]); %satır sayısı 133 sütun sayısı ölçekli  
>> imshow(B)
```



RESİM İYİLEŞTİRME (IMAGE ENHANCEMENT)

Görüntü restorasyonu konusu, bir görüntünün oluşumu esnasında oluşabilen veri kayıplarını veya bozulmaları azaltma veya tamamen yok etme konularını kapsayan; önemli konulardan birisidir.

- Görüntüdeki verim kaybı (görüntünün görünme derecesinin düşürülmesi) **gürültülerden** kaynaklanır. Gürültü pikselin gerçek değerindeki sapmadır.

- Gürültü**; hareket veya atmosferik kararsızlık nedeniyle meydana gelen bulanıklaşma veya resmi çekerken yanlış ışık etkisinden dolayı focus bulanıklaşması, kusursuz olmayan lenslerden kaynaklanan geometrik bozulma ve elektronik kaynaklardan gelen hatalar olarak verilebilir. Bunlar pikselin gerçek parlaklık değerine bir tesadüfi sayı ekler.

HİSTOGRAM KAVRAMI

Histogram bir resimdeki renk değerlerinin sayısını gösteren grafiktir. Histogram dengeleme veya eşitleme de bir resimdeki renk değerlerinin belli bir yerde kümelenmiş olmasından kaynaklanan, renk dağılımı bozukluğunu gidermek için kullanılan bir yöntemdir.

- Histogram matematiksel olarak aşağıdaki şekilde gösterilebilir.

- $h(r_k)=n_k$

r_k : k'nıncı parlaklık değeri

n_k : k nıncı parlaklık değerinin görüntüdeki sayısı

Örnek:

Aşağıda verilen kısmen karanlık sayılabilecek olan görüntünün histogram eşitleme yöntemi ile düzenlenmesi..

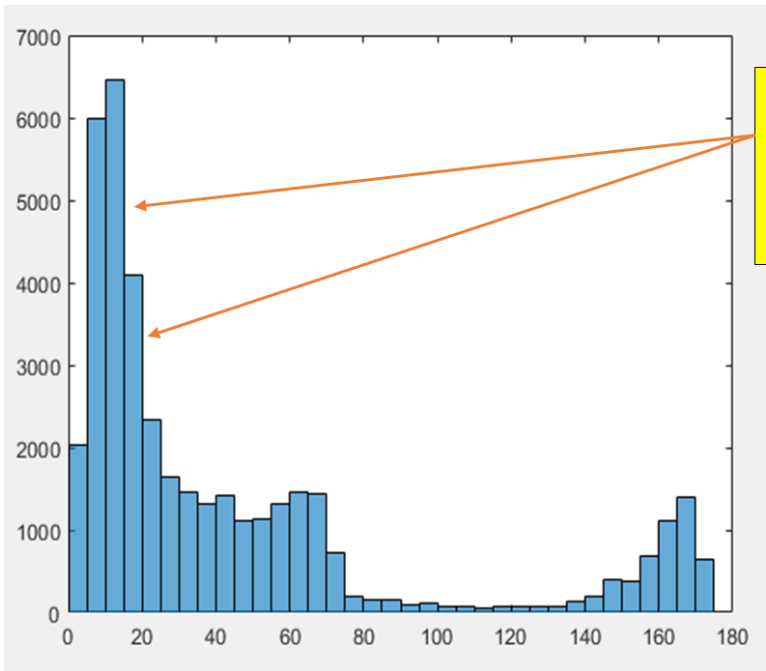


Histogram eşitleme işleminden önceki durum

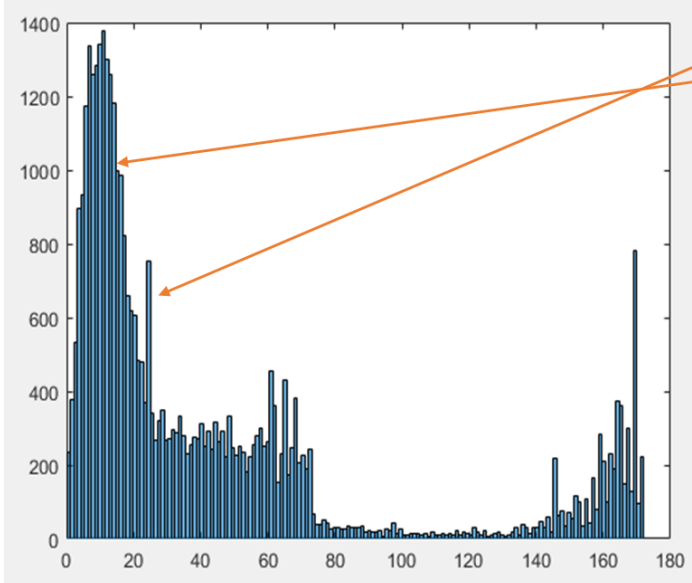


%histogram eşitleme (germe) işlemi

```
x=imread('headquarters-2K.png');  
figure  
imshow(x);  
x=rgb2gray(x);  
y=histogram(x);  
y.Values %Herbir çubuk bar(bin)'daki piksel sayısı  
[a,b]=max(y.Values)%En fazla piksele sahip parlaklık değeri  
aralığı bulunur.
```

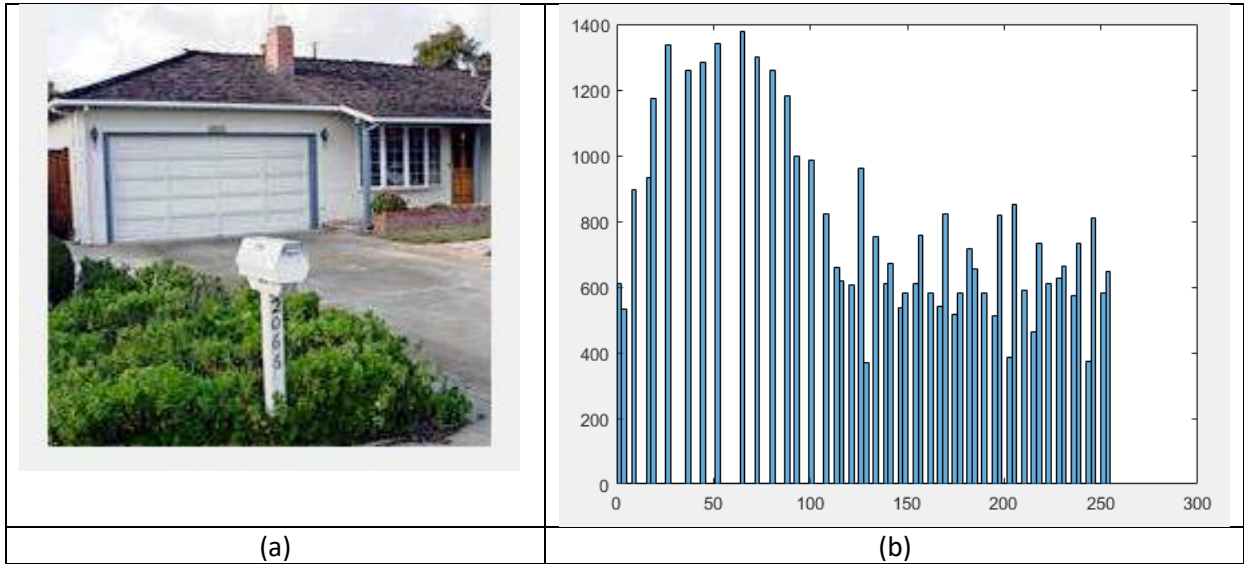


```
y=histogram(x,'NumBins',165);
```



Histogramı eşitleme işlemi:

```
J=histeq(x);  
imshow(J);  
histogram(J);
```



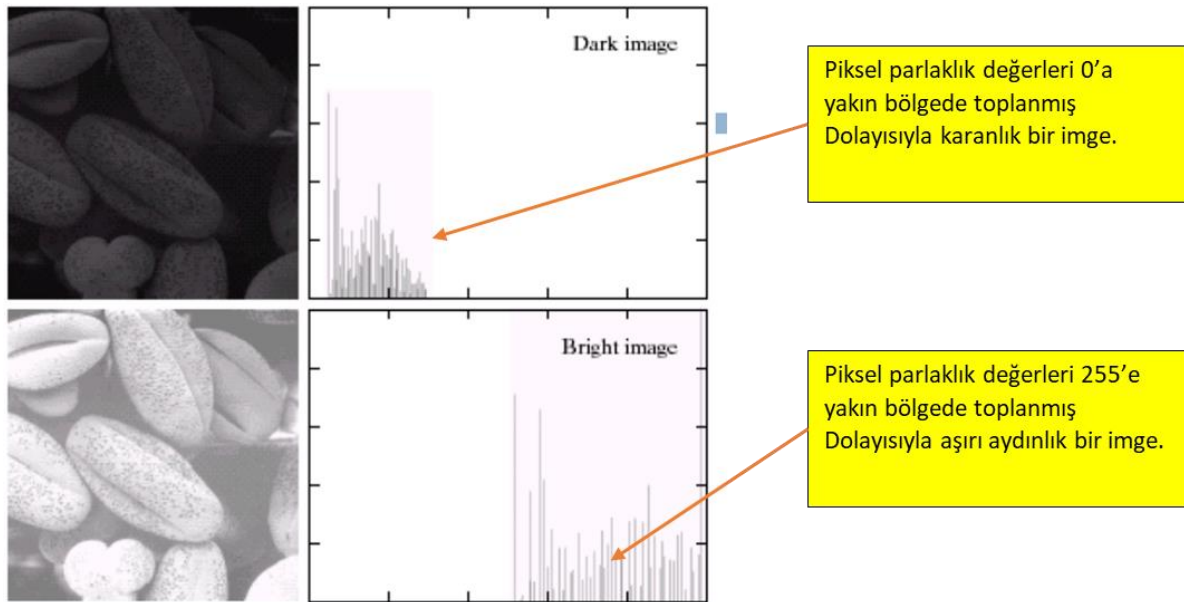
Histogram eşitleme işleminden sonraki durum;

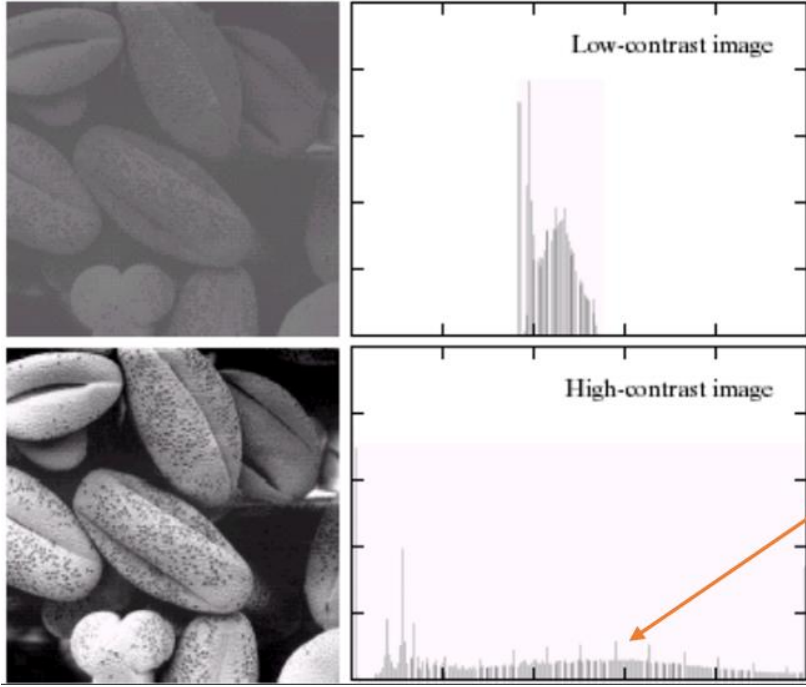
(a)Histogramı eşitlenmiş resim

(b)Histogram eşitlemeden sonra oluşan histogram.

Görüldüğü üzere histogram çubukları ilk histograma göre daha homojen dağılmış. Sonuç olarak resmin kalitesi artmıştır.

Bir başka örnek:





Histogramı eşitlenmiş imge.
Piksel parlaklık değeri homojen
dağılmış.

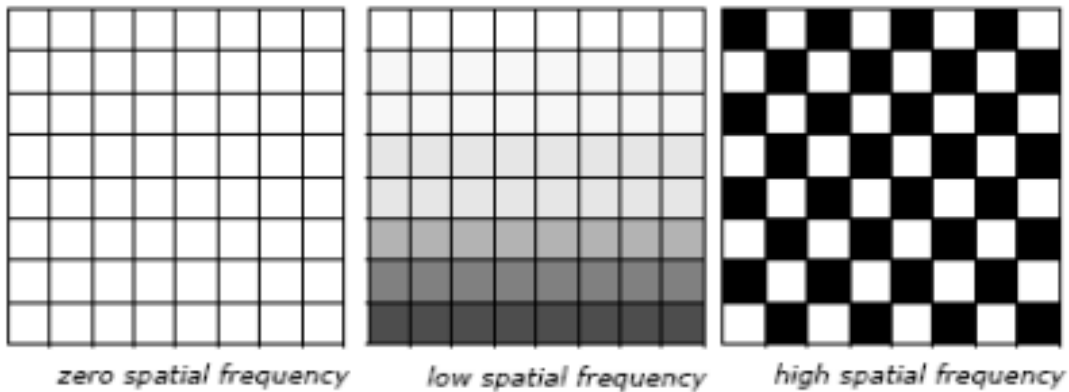
GÖRÜNTÜ FİLTRELEME

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:

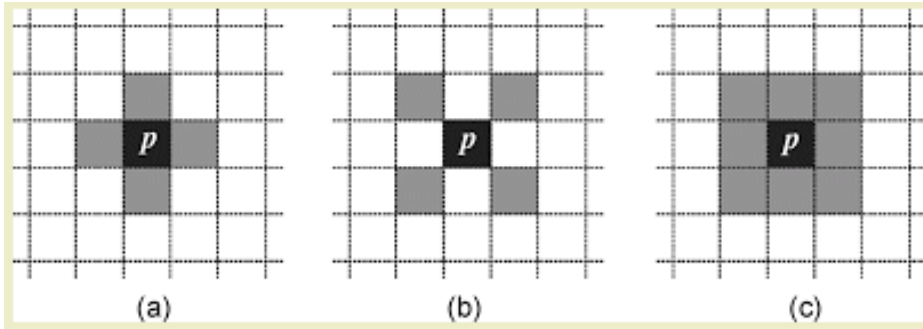




PİKSEL KOMŞULUKLARI :

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;

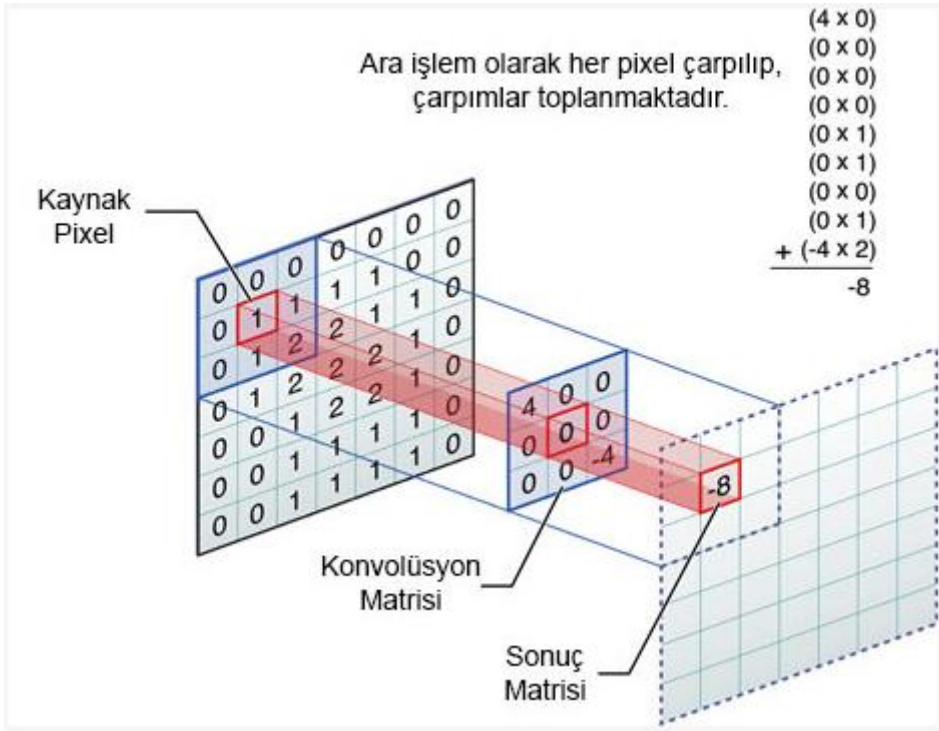
- w dizisinin en sağdaki noktasının, f dizisinin en soldaki noktasına gelecek şekilde konumlandırılarak işlem başlatılıp, her seferinde w dizisini bir adım sağa kaydırarak işlemler tekrarlanır.
- f dizisi ile w dizisinin elemanları üste örtüşmeyebilir. Bu durumda f dizisinin sağına ve soluna gerektiği kadar 0 doldurulur. (Zero padding)
- Böylelikle f dizisinin her elemanının w dizisi ile ilişkiye girmesi garanti edilmiş olur.

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

3	'full' convolution result 0 0 0 1 2 3 2 0 0 0 0
4	'same' convolution result 0 1 2 3 2 0 0

İki boyutlu korelasyon ve konvolüsyon:

İki boyutlu korelasyon ve konvolüsyon prensip olarak bir boyutluyla aynıdır. Konvolüsyon işleminde kullanılan filtre korelasyon işleminde kullanılanın 180° döndürülmüş halidir. 2 Boyutlu konvolüsyon işlemini gösteren şekil aşağıda gösterilmiştir. Bu defa filtre (maske, şablon, çekirdek) matrisinin tam merkezindeki piksel kaynak görüntü matrisinin sol üst köşesindeki piksele karşılık gelecek şekilde işlem yapılır. Filtre matrisinde boşta kalan pikseller için kaynak matriste sıfırlardan oluşan satır ve sütunlar eklenir (zero padding).



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

$x=\text{zeros}(7)$

$x =$

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



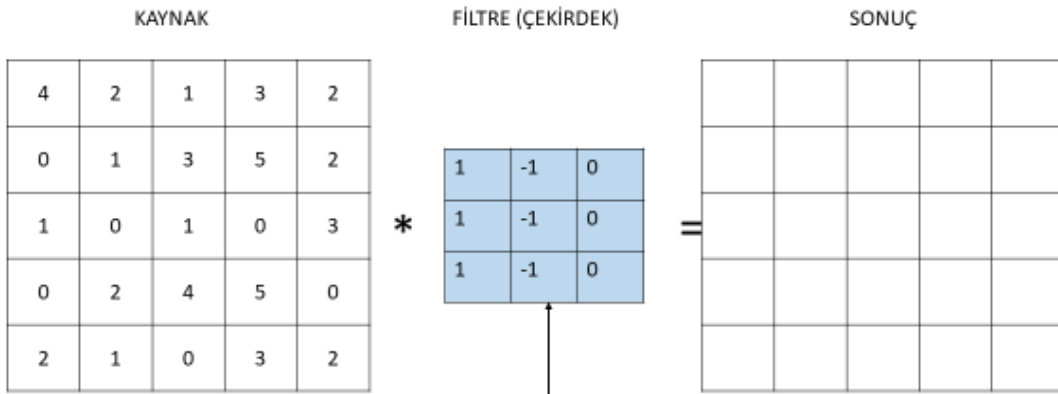
AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



```
>> 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
```

Konvolüsyon işlemi ayrıntılı örnek gösterimi:
1. adım:



Filtre Matrisi yukarıdaki gibidir ancak konvolüsyon işleminde filtre 180 derece çevrilir.



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



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

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



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



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	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	-4	

$$0x0 + (-1)x0 + 1x0 + 0x1 + (-1)x(3) + 1x2 + 0x3 + (-1)x5 + 1x2 = -4$$



8. adım:

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

-1	1	4	-4	-4

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

9. adım:

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

-1	1	4	-4	-4
-2				

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



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



10. adım:

0	0	0	0	0	0	0
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	-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	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	-4	-4
-2	2	3		

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



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



. . . 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

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

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