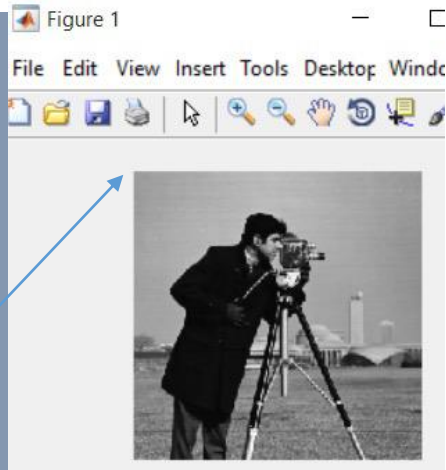


TEMEL GÖRÜNTÜ İŞLEME TEKNİKLERİ

Görüntü Özellikleri

```
clear all;
clc;
close all;
imgIkk = imread('cameraman.tif');
size(imgIkk)
whos imgIkk
%Çalışma alanındaki değişkenleri boyutları ve türleriyle listeleme
c1=impixel(imgIkk,2,5) %piksel renk değerlerini ver
imfinfo('cameraman.tif')
imshow (imgIkk)
```



```
ans =
    256    256

Name          Size          Bytes  Class
imgIkk        256x256        65536  uint8

c1 =
    153    153    153

ans =
```

```
Filename: 'D:\Programla
FileModDate: '04-Ara-2000
FileSize: 65240
Format: 'tif'
FormatVersion: []
Width: 256
Height: 256
BitDepth: 8
ColorType: 'grayscale'
FormatSignature: [77 77 0 42]
ByteOrder: 'big-endian'
NewSubFileType: 0
BitsPerSample: 8
Compression: 'PackBits'
```

Variables - imgIkk											
imgIkk											
256x256 uint8											
	1	2	3	4	5	6	7	8	9	10	11
1	156	159	158	155	158	156	159	158	157	158	158
2	160	154	157	158	157	159	158	158	158	160	155
3	156	159	158	155	158	156	159	158	157	158	158
4	160	154	157	158	157	159	158	158	158	160	155
5	156	153	155	159	159	155	156	155	155	157	155
6	155	155	155	157	156	159	152	158	156	158	152
7	156	153	157	156	153	155	154	155	157	156	155
8	159	159	156	158	156	159	157	161	162	157	157
9	158	155	158	154	156	160	162	155	159	161	156
10	155	154	157	158	160	160	159	160	158	161	160
11	154	157	157	157	156	155	150	154	150	158	161

workspace	
Name	Value
ans	1x1 struct
c1	[153 153 153]
imgIkk	256x256 uint8

Basit Görüntü İşlemleri

```
clear all;clc;close all;  
A=imread('D:\resler\Lena.tiff');  
AA=rgb2gray(A); % RGB görüntüsünü veya renk haritasını gri tonlamaya dönüştürür  
B=imbinarize (AA); % Eşiğe göre görüntüyü ikili görüntüye dönüştürün  
C=~B; % logical NOT  
figure;  
subplot(1,4,1); imshow(A); title('Resmin Orijinali');  
subplot(1,4,2); imshow(AA); title('Resmin Grisi')  
subplot(1,4,3); imshow(B); title('Siyah-Beyaz');  
subplot(1,4,4); imshow(C); title('Beyaz-Siyah');
```

Resmin Orijinali



Resmin Grisi



Siyah-Beyaz



Beyaz-Siyah



Basit Görüntü İşlemleri

```
clear all; clc; close all;  
imgllk = imread('cameraman.tif');  
%Görüntüyü dikey olarak döndürmek için;  
fp=imgllk(end:-1:1,:); % (a:k:b), a-ilk indeksi, b-sonuncuyu ve k-adım boyutunu belirtir.  
imshow(fp);
```



```
clear all; clc; close all;  
imgllk = imread('cameraman.tif');  
%Görüntüyü kesmek için;  
fc=imgllk(100:200,100:200);  
imshow(fc);
```



```
clear all; clc; close all;  
imgllk = imread('cameraman.tif');  
%Görüntüyü küçültmek için;  
fs=imgllk(1:4:end,1:4:end); %256x256 görüntü 64x64 haline gelir  
imshow(fs);
```

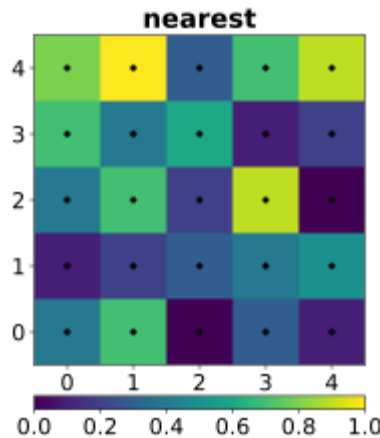


Geometrik İşlemler

- **Döndürme, oranlı büyütme, x ve y eksenlerinde değişik oranlarda boyutlandırma** işlemleri 2 aşama ile gerçekleştirilir.
- Eldeki piksel değerleri kullanılarak gerekli dönüşümler yapılır.
- Enterpolasyon kullanılarak dönüşüme uğramış resmin tamsayı piksel koordinatları bulunur.
- Enterpolasyon: İlk kez Uygulamalı Matematik biliminin bir alt kategorisi olan Sayısal Analiz yöntemlerinde tanımlanan ve elde varolan (bilinen) **değer noktalarından yola çıkarak bu noktalar arasında, farklı bir yerde ve değeri bilinmeyen bir noktadaki olası değeri bulmaya/tahmin etmeye yarayan yöntemlerin tümüne** verilen genel isimdir.
- Enterpolasyon genelde mühendislik ve deneylere/ölçümlere dayalı benzeri bilim dallarında, toplanan verilerin bir fonksiyon eğrisine uydurulması amacıyla kullanılmaktadır.

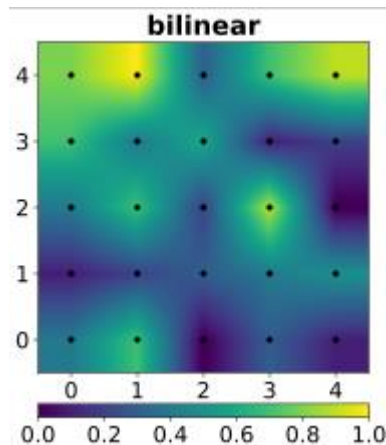
Geometrik İşlemler

- **Nearest Enterpolasyon**
- En yakın komşu enterpolasyonu: Çıkış görüntüsündeki her pikselin değeri, giriş görüntüsündeki en yakın pikselin değeri alınarak belirlenir. Bu yöntem hızlı ve basittir ancak kenarların pürüzlü olmasına ve ayrıntı kaybına neden olabilir.



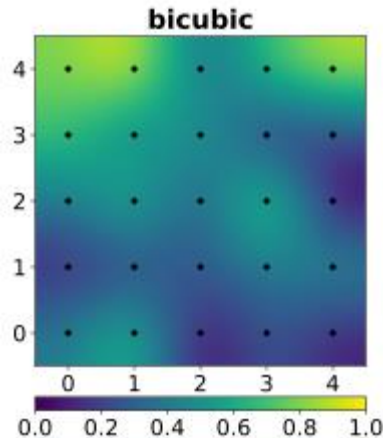
Geometrik İşlemler

- **Bilinear Enterpolasyon**
- Çift doğrusal enterpolasyon: Çıkış görüntüsündeki bir pikselin değerini, giriş görüntüsündeki en yakın dört pikselin değerlerinin ağırlıklı ortalaması olarak hesaplar. En yakın komşu ve doğrusal enterpolasyona göre daha karmaşık bir yöntem olup, renkler arasında daha yumuşak geçişlerle daha kaliteli sonuçlar üretir.

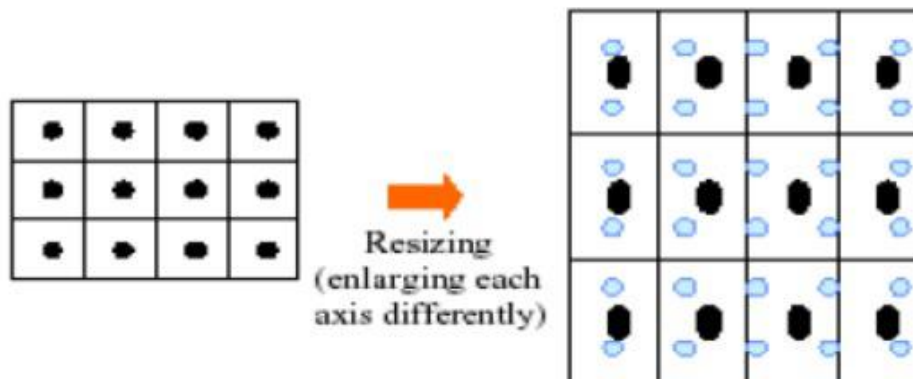
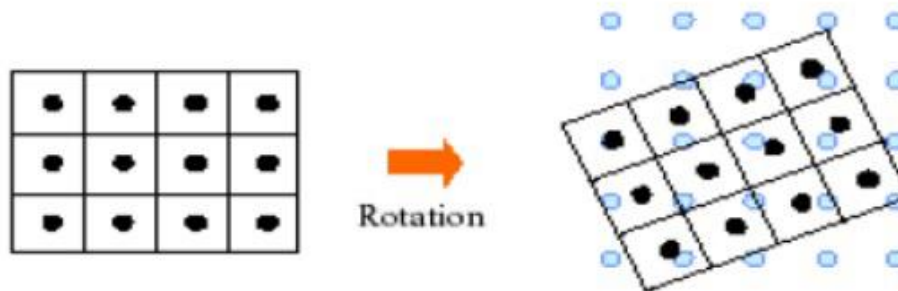


Geometrik İşlemler

- **Bicubic Enterpolasyon**
- Bikubik enterpolasyon: Çıkış görüntüsündeki bir pikselin değerini, giriş görüntüsündeki en yakın 16 (4x4) pikselin değerlerinin ağırlıklı ortalaması olarak hesaplar.



Geometrik İşlemler

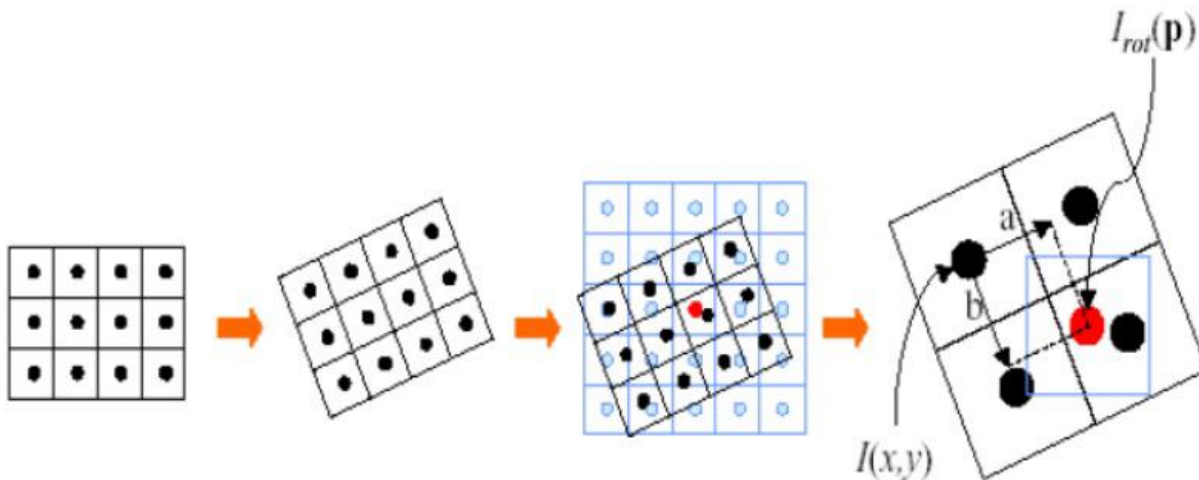


Döndürme

- Yeni piksel değeri döndürme matrisi yardımı ile elde edilir.
- Yeni tamsayı değeri örneğin bilinear enterpolasyon kullanılarak elde edilir.

$$\mathbf{p}_{rot} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \mathbf{p}$$

$$I_{rot}(\mathbf{p}) = \begin{bmatrix} 1-b & b \end{bmatrix} \begin{bmatrix} I(x,y) & I(x+1,y) \\ I(x,y+1) & I(x+1,y+1) \end{bmatrix} \begin{bmatrix} 1-a \\ a \end{bmatrix}$$



Matlab ile Image Döndürme

- Matlab ile kolay bir şekilde resim dosyasını istediğimiz açıyla döndürebiliriz.
- **`I2=imrotate(I1,açı,yöntem);` şeklinde yazabiliriz.**
- Burada;
- `I1`: döndürülecek resim matris değerleridir.
- `Açı`: saat yönünün tersine kaç derece döndürüleceğini belirtir.
- `Yöntem`: Döndürme işlemi sonrası imgelerin yeni piksel değerlerinin hangi ara değerlendirme yöntemi ile belirleneceğini belirtmede kullanılır. 'nearest', 'bilinear', 'bicubic' değerlerini alabilir. Belirtilmemesi durumunda varsayılan yöntem nearest'dir.
- Örnek: **`Im=imrotate(I,65,bicubic);`**

Matlab ile Image Döndürme

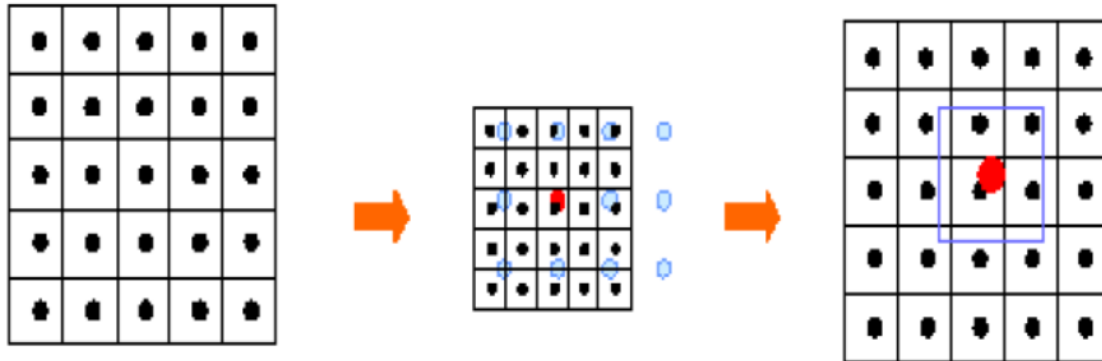
```
I=imread('cameraman.tif');  
I2 = imrotate(I, 30, 'nearest');  
figure, imshow(I)  
figure, imshow(I2)
```



Boyutlandırma

- İlk önce, boyutlandırma matrisinden geçirilerek yeni piksel değeri elde edilir.
- Enterpolasyon yardımı ile yeni tamsayı piksel elde edilir.
- Bu enterpolasyonda, bilinen komşu pikseller kullanılabilir.

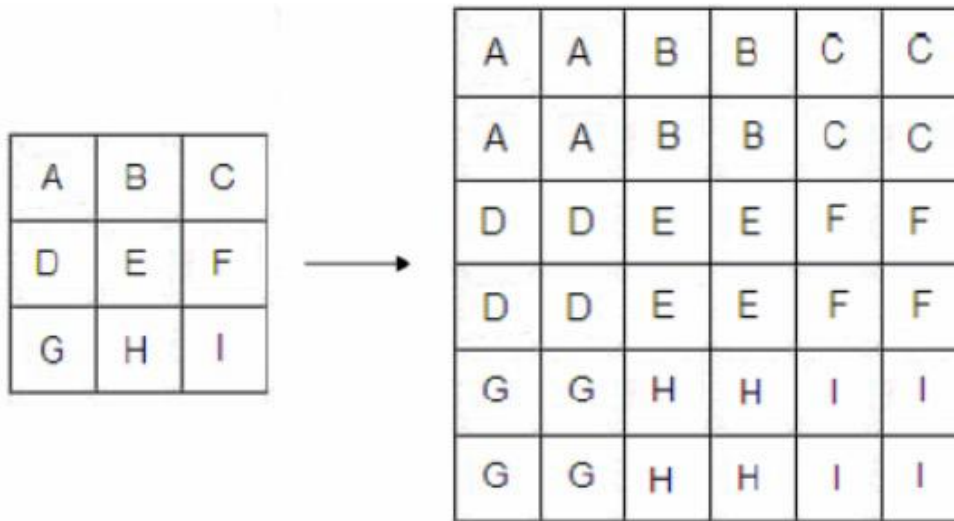
$$\mathbf{p}_{scaled} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \mathbf{p}$$



Matlab ile Boyutlandırma

- Matlab ile boyut değiştirme için “imresize” işlevi kullanılabilmektedir.
- **`Is=imresize(I,oran,yöntem);`**
- oran: giriş imgesinin boyutunun değişme oranını verir.
- $\text{oran} > 1$ (büyütme), $\text{oran} < 1$ (küçültme).
- yöntem: boyut değiştirmede kullanılacak ara değerlendirme yöntemi.
- Örnek: **`Is=imresize(I,0.97, 'bicubic');`**

Yakınlaştırma-Zoom In



3x3'lük bir imgenin yakınlaştırma öncesi ve sonrasında piksel değerleri harflerle gösterilmiştir. Burada yapılan sadece ilgili piksel değerinin imge boyutunu iki katına çıkaracak şekilde yeni imgeye kopyalanmasıdır. (nearest neighbor interpolation)

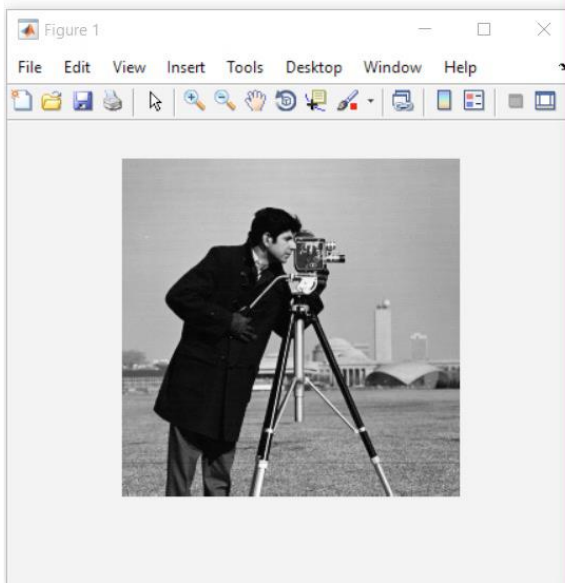
Yakınlaştırma-Zoom In

A	B	C
D	E	F
G	H	I

A	$\frac{A+B}{2}$	B	$\frac{B+C}{2}$	C
$\frac{A+D}{2}$	$\frac{A+B+D+E}{4}$	$\frac{B+E}{2}$	$\frac{B+C+E+F}{4}$	$\frac{C+F}{2}$
D	$\frac{D+E}{2}$	E	$\frac{E+F}{2}$	F
$\frac{D+G}{2}$	$\frac{D+E+G+H}{4}$	$\frac{E+H}{2}$	$\frac{E+F+H+I}{4}$	$\frac{F+I}{2}$
G	$\frac{G+H}{2}$	H	$\frac{H+I}{2}$	I

Matlab ile Yakınlaştırma-Zoom In

```
clear all; clc; close all;  
I=imread('cameraman.tif');  
Is=imresize(I,1.5,'bicubic');  
imshow(I),figure,imshow(Is);
```



Uzaklaştırma-Zoom Out

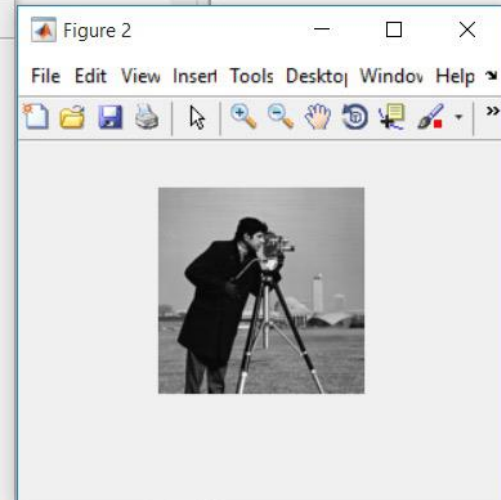
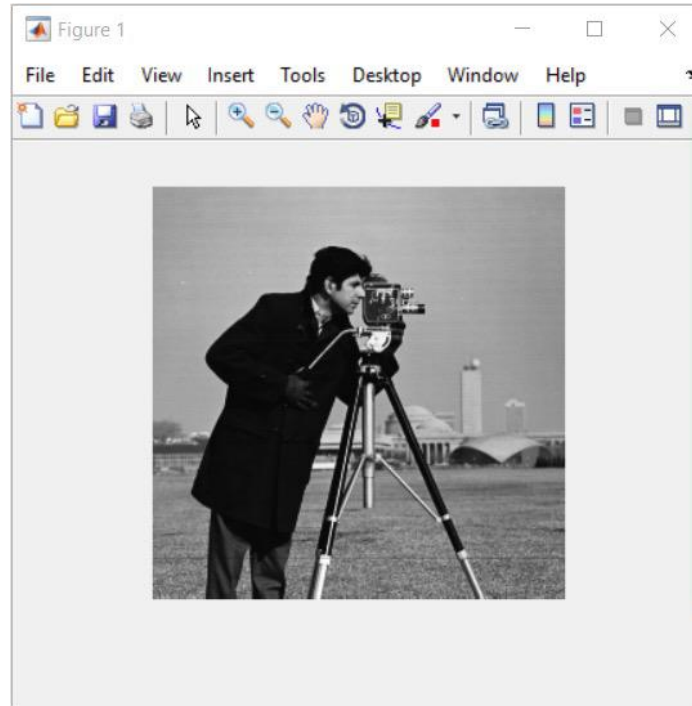
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

$\frac{A+B+E+F}{4}$	$\frac{C+D+G+H}{4}$
$\frac{I+J+M+N}{4}$	$\frac{K+L+O+P}{4}$

Matlab ile Uzaklaştırma-Zoom Out

- Matlab ile boyut değiştirme için “imresize” adındaki işlev kullanılabilir. **`Is=imresize(I,oran,yöntem);`**
- oran: giriş imgesinin boyutunun değişme oranını verir.
- $\text{oran} > 1$ (büyütme), $\text{oran} < 1$ (küçültme).
- yöntem: boyut değiştirmede kullanılacak ara değerlendirme yöntemi.

```
clear all; clc; close all;  
I=imread('cameraman.tif');  
Is=imresize(I,0.5,'bicubic');  
imshow(I),figure,imshow(Is);
```



Boyutlandırma- Matlab Örnek

%bir görüntünün büyütülmesi ve küçültülmesi

```
I = imread('C:\Lena.tiff');
```

```
J = imresize(I, 0.5);
```

```
figure, imshow(I), figure, imshow(J)
```

%en yakın komşu piksel değeri yöntemine göre bir görüntünün büyütülmesi ve küçültülmesi

```
I = imread('C:\Lena.tiff');
```

```
J = imresize(I, 0.5, 'nearest');
```

```
figure, imshow(I), figure, imshow(J)
```

%bilineer interpolasyon yöntemine göre bir görüntünün büyütülmesi ve küçültülmesi

```
I = imread('C:\Lena.tiff');
```

```
J = imresize(I, 0.5, 'bilinear');
```

```
figure, imshow(I), figure, imshow(J)
```

%bicubic interpolasyon yöntemine göre bir görüntünün büyütülmesi ve küçültülmesi

```
I = imread('C:\Lena.tiff');
```

```
J = imresize(I, 0.5, 'bicubic');
```

```
figure, imshow(I), figure, imshow(J)
```

Görüntü Öteleme

Öteleme işlemi, yatay ve düşey eksenlerde belirlenen piksel miktarı kadar imgenin yatay ve düşey ekseninde kaydırılması ile gerçekleştirilir.

30 piksellik yatay, 10 piksellik düşey öteleme için orjinal ve ötelenmiş "Akiyo" imgesi aşağıda verilmektedir.



Orjinal imge

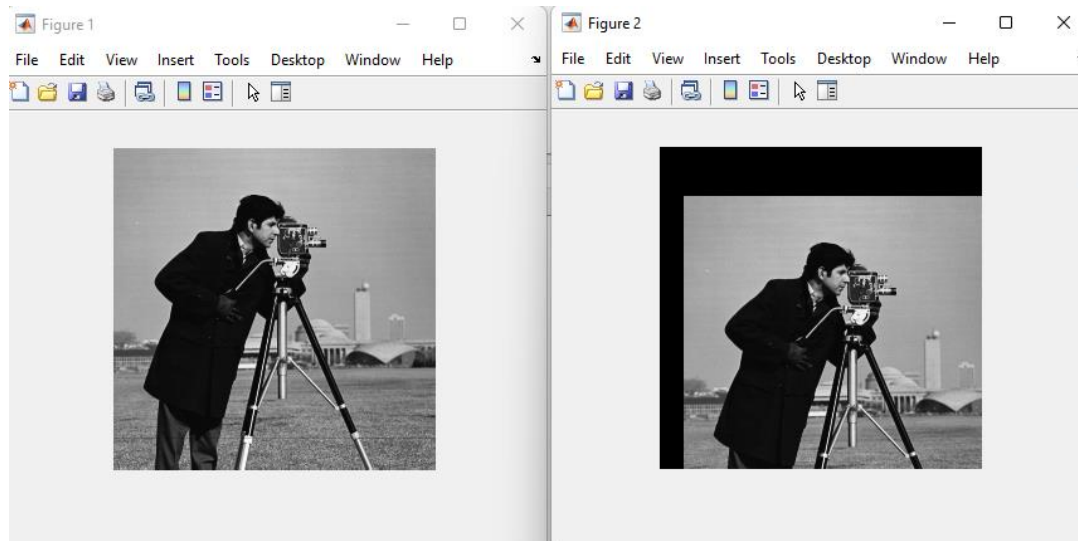


Ötelenmiş imge

Matlab ile Öteleme-Shift

```
function[B]=my_otele(A,n1,n2)
[w,h]=size(A);
B=zeros(w,h);
for i=n1:w
    for j=n2:h
        B(i,j)=A(i-n1+1,j-n2+1);
    end
end
```

```
clear all;close all;clc;
D=imread('cameraman.tif');
K=my_otele(D,40,20);
K=uint8(K);
imshow(D);figure;imshow(K);
```



Matlab ile Öteleme-Shift

```
clc;  
clear;  
close all;  
img = imread('peppers.png');  
figure,  
subplot(1,2,1);  
imshow(img);title('Orijinal');  
s = size(img);  
shift = 0.3; %relative shift  
img = img(:,mod(round((1:s(2))+ s(2)*shift), s(2))+1,:);  
subplot(1,2,2);  
imshow(img);title('Kaydırılmış');
```

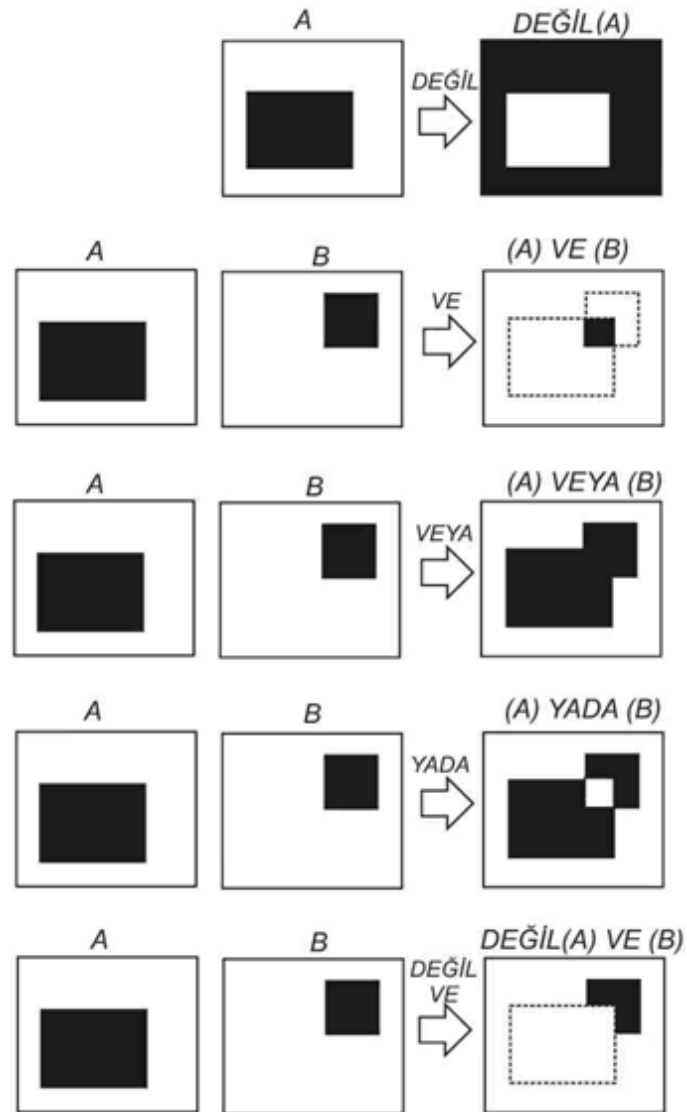
Orijinal



Kaydırılmış



Görüntüde (İmge) Mantıksal İşlemler



Görüntüde (İmge) Aritmetik İşlemler

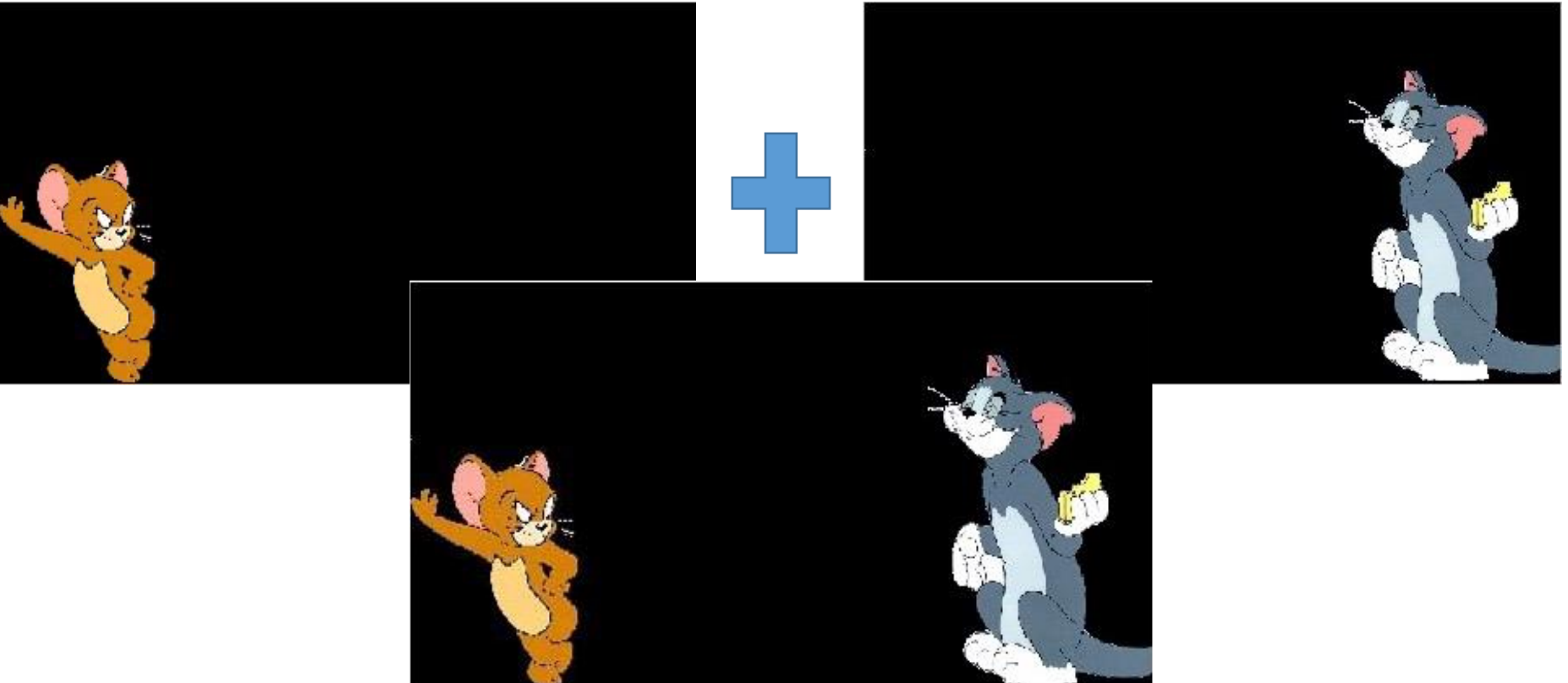
- **Aritmetik operatörler** görüntüye **piksel piksel uygulanır**.
- Yani çıkış görüntüsündeki bir pikselin değeri sadece giriş resminde o piksele karşılık gelen pikselin değerine bağlıdır. Bu yüzden çıkış ve giriş görüntüleri aynı uzunlukta olmak zorundadır.
- Giriş görüntülerinden biri **sabit bir sayı da olabilir** (bir görüntüye sabit bir offset değeri eklenmesinde olduğu gibi).
- Aritmetik operatörlerin en önemli avantajı işlemin oldukça **hızlı** olmasıdır.
- **Toplama** operatörü **kirlilik azaltmada**,
- **Çıkarma** operatörü örneğin medikal görüntülerde bulunan **statik arka planların çıkarılmasında**,
- **Çarpma ve bölme** operatörleri ise **renkli görüntünün grinin seviyelerine çevirmede** kullanılmaktadır.
- Aritmetik operatörler sonucunda **piksellerin koordinatları değişmemekte** sadece **renk** değerleri değişmektedir.

Toplama	$Q(i, j) = P_1(i, j) + P_2(i, j)$	$Q(i, j) = P_1(i, j) + C$
Çıkartma	$Q(i, j) = P_1(i, j) - P_2(i, j)$	$Q(i, j) = P_1(i, j) - C$
Çarpma	$Q(i, j) = P_1(i, j) \times P_2(i, j)$	$Q(i, j) = P_1(i, j) \times C$
Bölme	$Q(i, j) = P_1(i, j) \div P_2(i, j)$	$Q(i, j) = P_1(i, j) \div C$
Harmanlama	$Q(i, j) = X \times P_1(i, j) + (1 - X) \times P_2(i, j)$	

İki Resim Arasında Aritmetik İşlemler

Toplama

```
close all;clc;clear all;  
A=imread('D:\resler\tommy1.jpg');  
B=imread('D:\resler\jerry1.jpg');  
c=imadd(A,B);  
figure,imshow(c);
```



İki Resim Arasında Aritmetik İşlemler Toplama

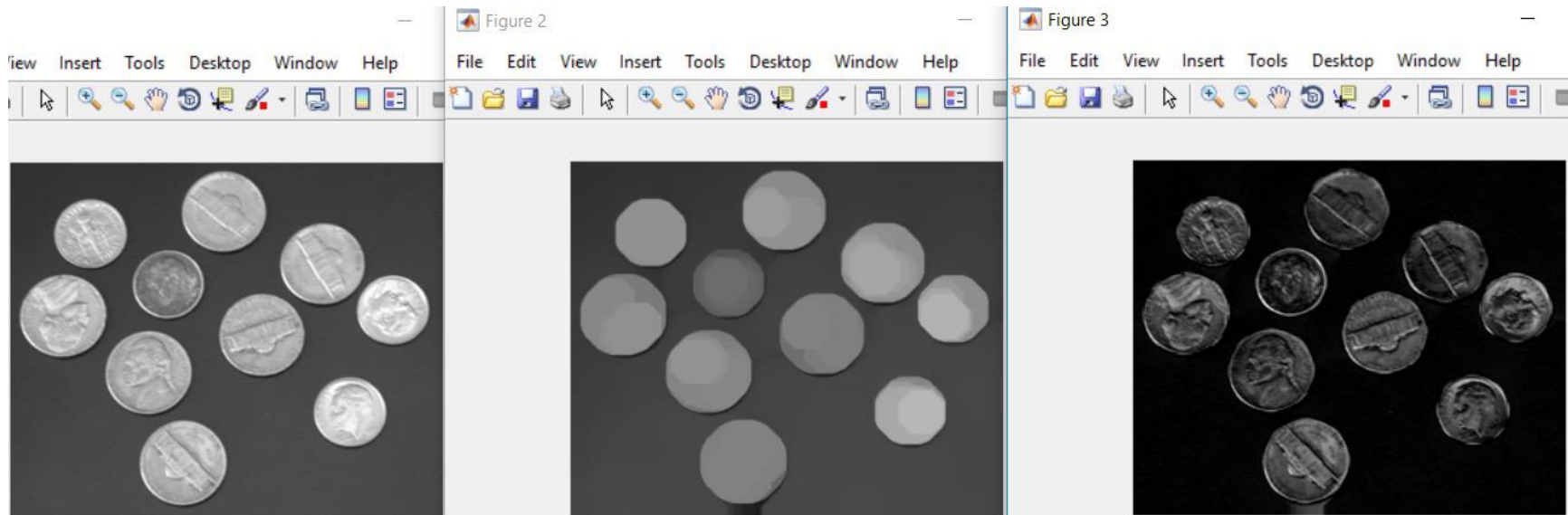
```
clc;  
clear;  
a = imread('rice.png');  
b = imread('cameraman.tif');  
c=imadd(a,b);%c=a+b;  
figure,imshow(a),title('Resim1');  
figure,imshow(b),title('Resim2');  
figure,imshow(c),title('Resim Toplama');
```



İki Resim Arasında Aritmetik İşlemler

Çıkarma

```
close all;clc;clear all;  
I = imread('D:\resler\coins.png');  
background = imopen(I,strel('disk',15));  
Ip = imsubtract(I,background);  
imshow(I);  
figure; imshow(background);  
figure; imshow(Ip,[])
```



«strel» komutu

strel: Morfolojik işlemlerde kullanılan yapısal filtre elemanıdır. Morfolojik işlemleri hangi şekil ve parametrelerle uygulayacağımızı strel ile belirleriz.

► Kullanım şekli; **SE = strel (filtreleme şekli, parametre)**

► Örnek;

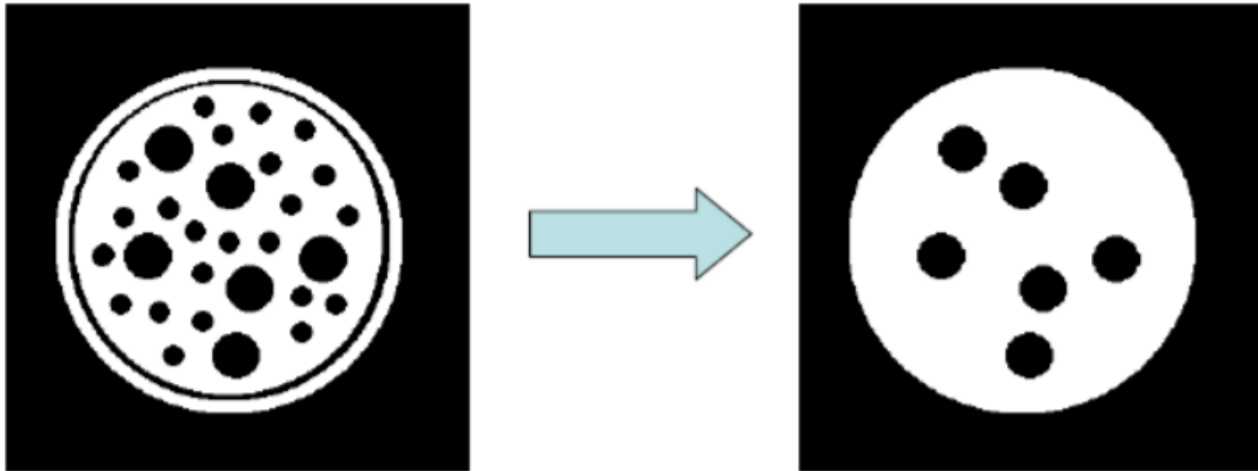
SE =strel('disk', R) R; yarıçap

SE = strel('square', L) L; karenin bir kenarının uzunluğu

Yapısal Filtre Elemanı: İmge üzerinde yapacağımız işleme ve yapmak istediğimiz uygulamaya göre istenilen boyutlarda ve istenilen şekilde hazırlanmış küçük ikilik bir imgedir. Yapısal eleman farklı geometrik şekillerden herhangi biri olabilir. En çok kullanılanları kare, dikdörtgen ve dairedir.

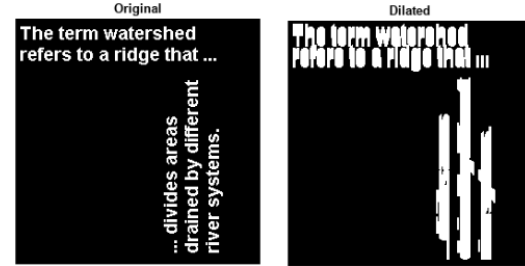
Morfolojik İşlemler

- Canlıların şekil ve yapıları ile ilgilenen dalına **morfoloji** (biçim) adı verilmektedir.
- Matematiksel **morfoloji**, temeli küme işlemlerine dayanan, imgedeki sınırlar (borders), iskelet (skeleton) gibi yapıların tanımlanması ve çıkartılması, gürültü giderimi, bölütleme, süzgeçleme, inceltme (thinning), budama (pruning) gibi ön/son işlem olarak da sıkça kullanılırlar.
- **Morfoloji** Binary ve gri resimlerde kullanılır. İmge içerisinden çıkarmak veya ayırt etmek istediğimiz nesneleri, diğer bölgelerden ayırt etmek veya çıkarmak için kullanılmaktadır.

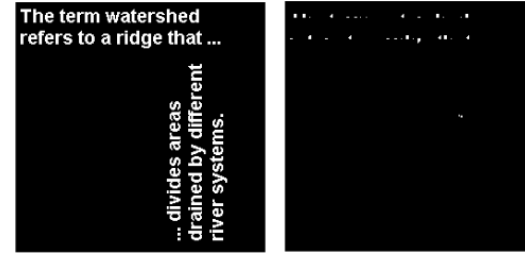


Morfolojik İşlemler

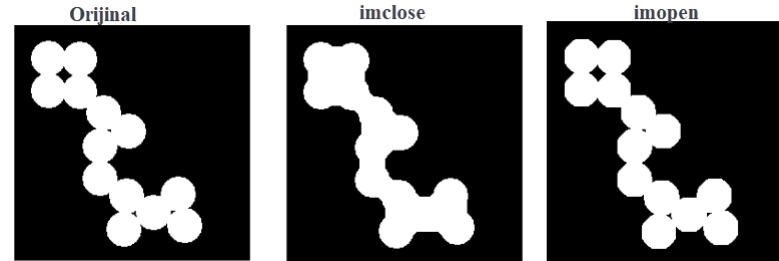
imdilate: genişleme işlemini yapar.



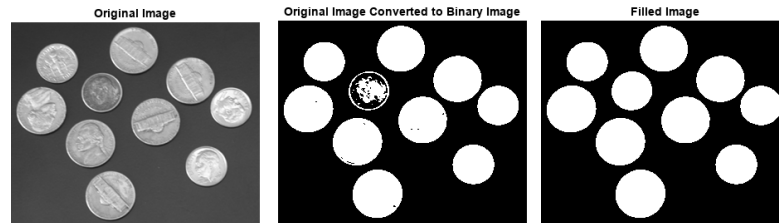
imerode: aşındırma işlemi yapar.



imclose, imopen: imdilate ve imerode işlemlerinin birlikte kullanılması ile oluşan filtrelerdir. Uygulama mantıkları benzerdir.



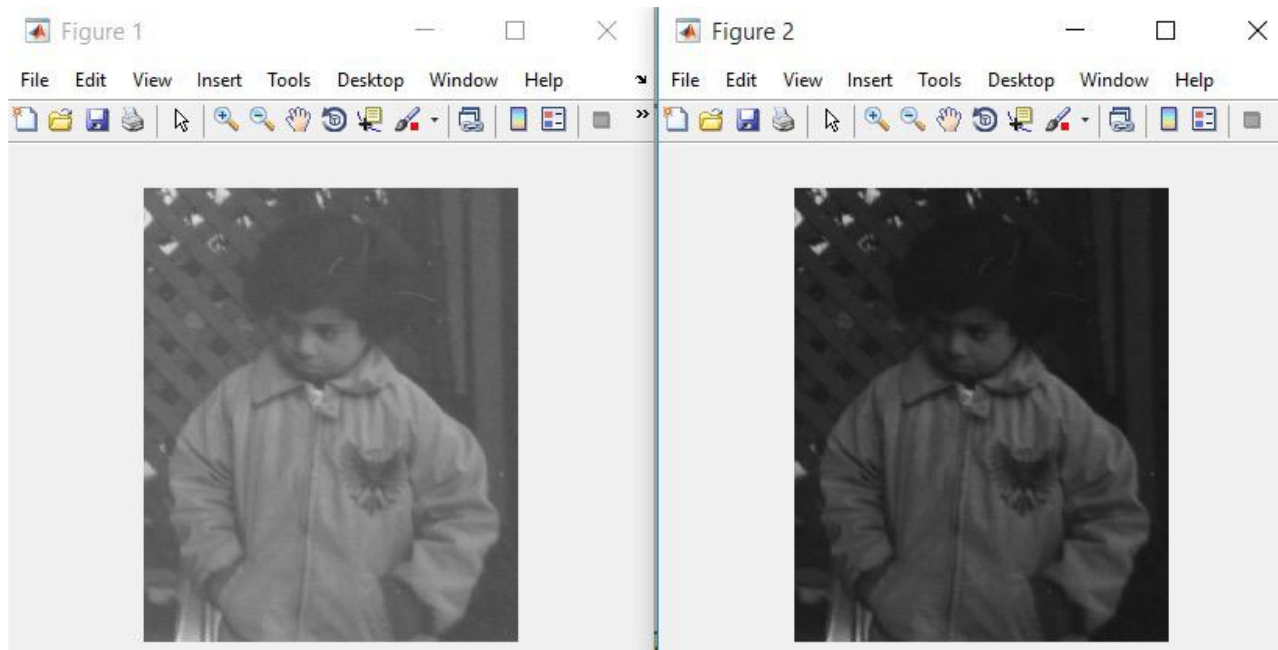
imfill: gri seviyede veya binary modda açıklıkları doldurur.



İki Resim Arasında Aritmetik İşlemler

Çarpma

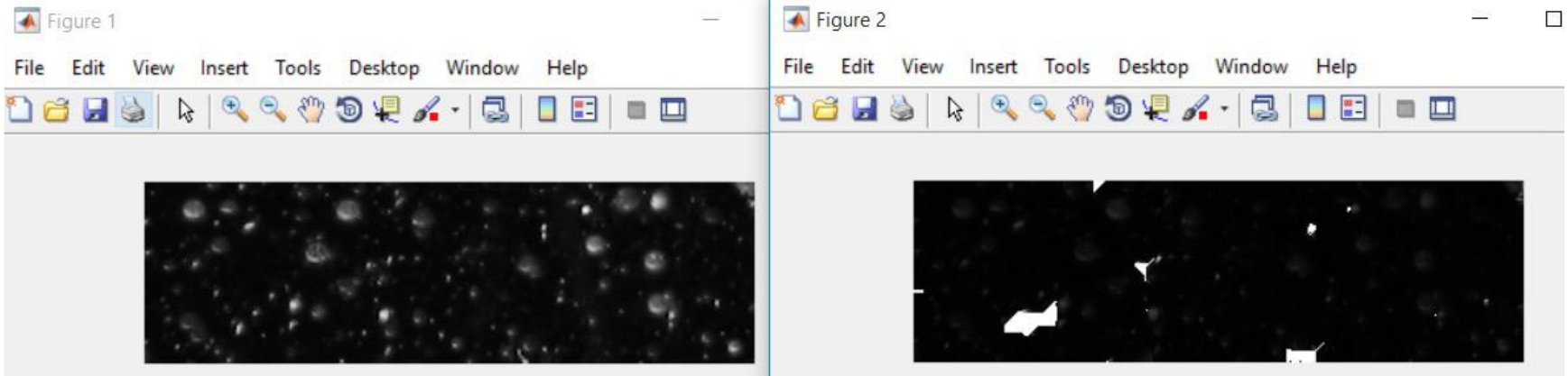
```
clear all;clc;close all;  
I = imread('D:\resler\pout.png');  
I16 = uint16(I);  
J = immultiply(I16,I16);  
imshow(I), figure, imshow(J)
```



İki Resim Arasında Aritmetik İşlemler

Bölme

```
clear all;clc;close all;  
I = imread('snowflakes.png');  
background = imopen(I,strel('disk', 5));  
Ip = imdivide(I,background);  
imshow(I), figure, imshow(Ip);
```



Görüntüde (İmge) Aritmetik İşlemler

```
clear all;close all;clc;  
yres=imread('D:\resler\trui.png');  
yres2=imadd(yres,80);  
yres3=imsubtract(yres,60);  
yres4=immultiply(yres,2);  
yres5=imdivide(yres,2);  
figure;  
subplot(1,5,1); imshow(yres); title('Resmin Orjinali');  
subplot(1,5,2); imshow(yres2); title('Resme Ekleme')  
subplot(1,5,3); imshow(yres3); title('Resimden Çıkarma');  
subplot(1,5,4); imshow(yres4); title('Resmi Çarpma');  
subplot(1,5,5); imshow(yres5); title('Resmi Bölme');
```

Resmin Orjinali



Resme Ekleme



Resimden Çıkarma



Resmi Çarpma



Resmi Bölme



Not: Resim piksellerine ilave edilen her değer resmin daha açık renkte görünmesine neden olur.

Not: Resim piksellerinden çıkarılan her değer resmin daha koyu renkte görünmesine neden olur.

Görüntüde (İmge) Aritmetik İşlemler

Noktasal operasyonlardan olan aritmetiksel işlemler;

$$y=f(x)$$

gibi bir basit fonksiyonun, görüntüdeki her bir gri seviye değerine uygulanmasından ibarettir. Gri seviye resimde genellikle 0,...255 seviye değerleri arasında çalışıldığından, yukarıdaki fonksiyon bu değerler arasında geçerli olmalıdır. Bu basit fonksiyon ;

Bir sabit değeri (c) her bir piksele ekler veya çıkartır ($Y=x \pm c$)

Veya her bir pikseli bir sabit değeri (c) ile çarpar ($y= x.c$)

İşlem sonuçlarının tamsayıya yuvarlatılması yapılır.

Ayrıca 0...255 arasında çalışıldığında kırpma yapılır.

$$y > 255 \quad \text{ise} \quad y=255$$

$$y < 0 \quad \text{ise} \quad y=0 \quad \text{yapılır.}$$

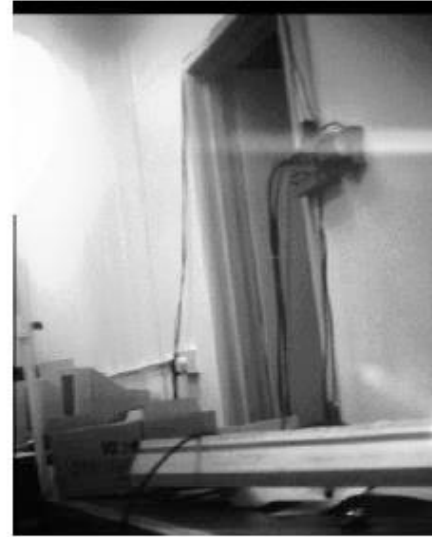
Görüntü (İmge) Pekiştirme

İmge Pekiştirme

İmge pekiştirmede amaçlanan çeşitli nedenlerden dolayı kullanışsız olan (yeterli bilgi vermeyen) imgenin işlenerek görsel veya işlevsel anlamda kullanılabilir bir imge elde etmektir.



Pozlandırma süresinden dolayı
oluşan karanlık imge



Pekiştirme işleminden sonra
elde edilen imge

Karşıtlık (Kontrast-Zıtlık) Ayarı



Parlaklık değişimi imgenin ışıklılık değerlerinin belirli bir sayıyla çarpılması ile gerçekleştirilir.

$$g(x, y) = af(x, y)$$

$a > 1$ ise karşıtlık arttırılır

$a < 1$ ise karşıtlık azaltılır



Karşıtlık (Kontrast-Zıtlık) Ayarı

```
clear all; close all;  
imgllk = imread('cameraman.tif');  
imgKar1 = imgllk*1.5;  
imgKar2 = imgllk*2;  
imgKarAz = imgllk*0.5;  
figure;  
subplot(1,4,1); imshow(imgllk); title('Resmin Orjinali');  
subplot(1,4,2); imshow(imgKar1); title('Contrast Arttı1');  
subplot(1,4,3); imshow(imgKar2); title('Contrast Arttı2');  
subplot(1,4,4); imshow(imgKarAz); title('Contrast Azaldı');
```

Resmin Orjinali



Contrast Arttı1



Contrast Arttı2



Contrast Azaldı



Parlaklık Ayarı

Parlaklık deęiřimi imgenin ıřıklılık deęerlerinin belirli bir sayıyla toplanması veya ıkarılması ile gerekleřtirilir. $f(x,y)$ orjinal imgeyi gstermek zere, parlaklık ayarı yapılmıř $g(x,y)$ ařaęıdaki řekilde elde edilir.

$$g(x, y) = f(x, y) + b$$

$b > 0$ ise parlaklık artırılır

$b < 0$ ise parlaklık azaltılır



Orjinal



$b = -50$



$b = +50$

Parlaklık Ayarı

```
clear all; close all;  
imgIkk = imread('cameraman.tif');  
imgPar = imgIkk+60;  
imgMat = imgIkk-60;  
figure;  
subplot(1,3,1); imshow(imgIkk); title('Resmin Orjinali');  
subplot(1,3,2); imshow(imgPar); title('Parlak Resim');  
subplot(1,3,3); imshow(imgMat); title('Mat-Koyu Resim');
```

Resmin Orjinali



Parlak Resim

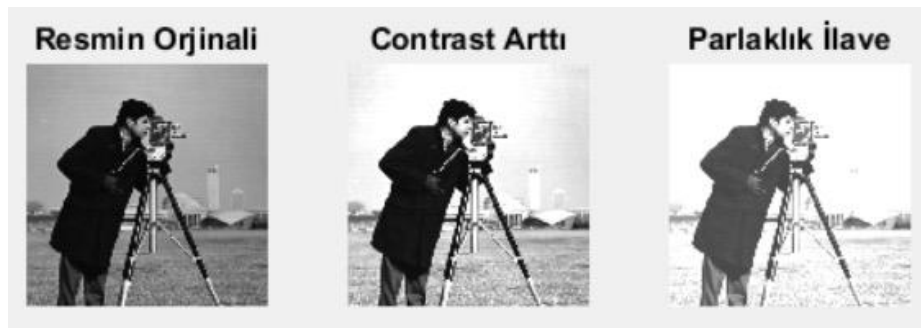


Mat-Koyu Resim



Karşıtlık (Kontrast) ve Parlaklık Ayarı

```
clear all; close all;  
imgIkk = imread('cameraman.tif');  
imgKar1 = imgIkk*1.5;  
imgPar2 = imgKar1+60;  
figure;  
subplot(1,3,1); imshow(imgIkk); title('Resmin Orjinali');  
subplot(1,3,2); imshow(imgKar1); title('Contrast Arttı');  
subplot(1,3,3); imshow(imgPar2); title('Parlaklık İlave');
```



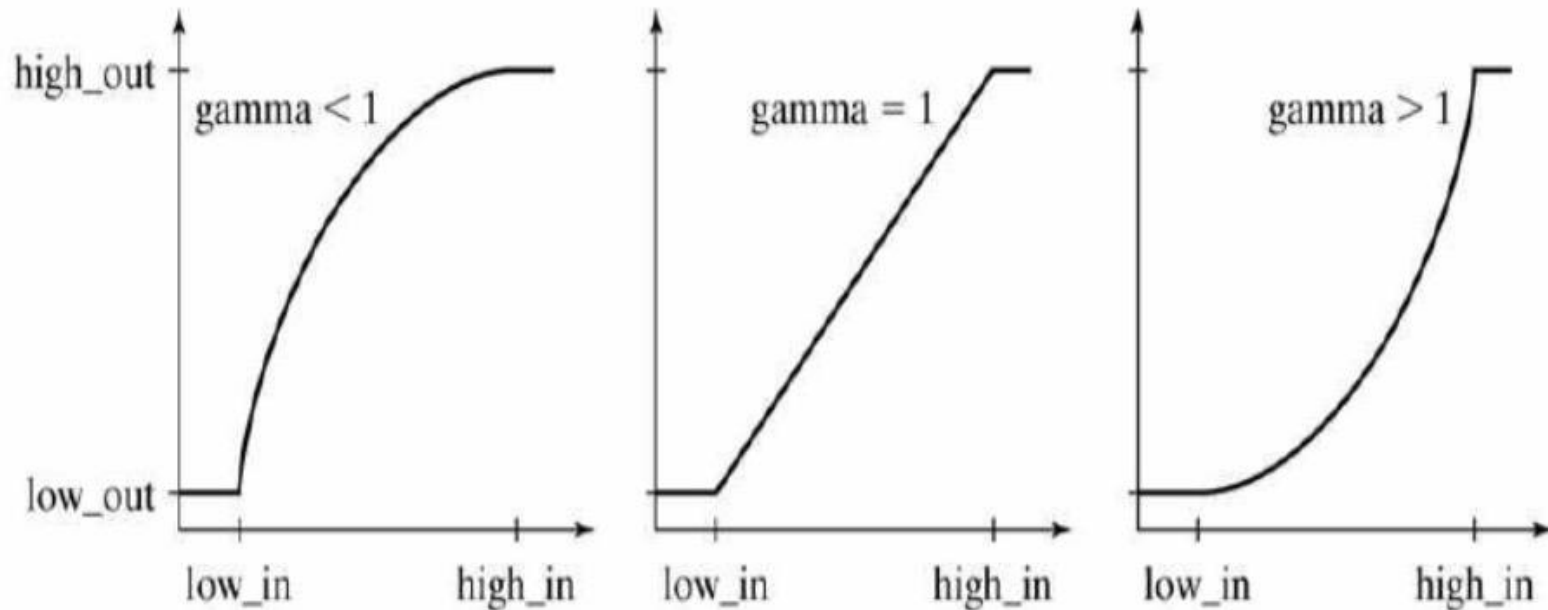
```
clear all; close all;  
imgIkk = imread('cameraman.tif');  
imgKarPar = (imgIkk*1.5)+60;  
figure;  
subplot(1,2,1); imshow(imgIkk); title('Resmin Orjinali');  
subplot(1,2,2); imshow(imgKarPar); title('Cont+Par Arttı');
```



«imadjust» Fonksiyonu

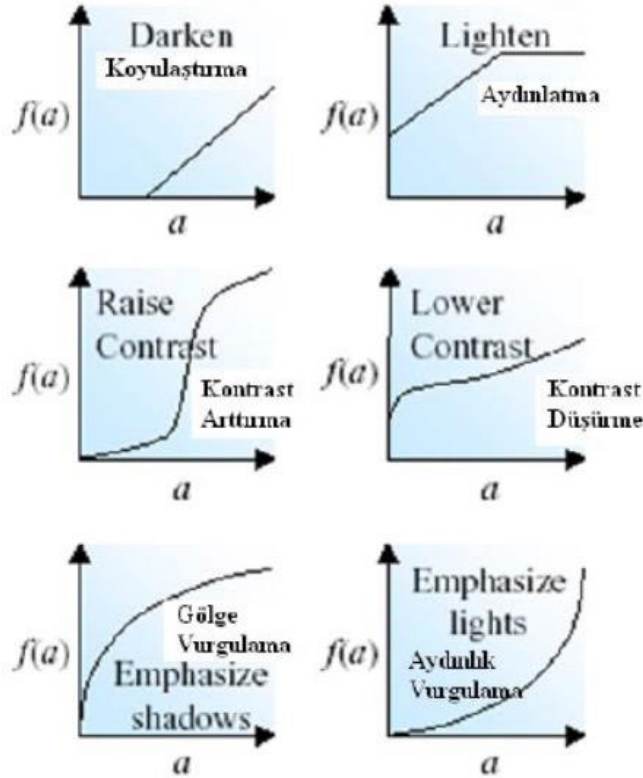
Temel parlaklık işleme fonksiyonudur. Gri skala görüntülerde parlaklık seviyesi dönüşümü yapar. Giriş image'ı f uint8,uint16,double olabilir. g çıkışı da aynı formattadır.

```
>> g= imadjust (f, [low_in high_in],[low_out high_out]), gamma
```



«imadjust» Fonksiyonu

En basit görüntü işlemleri , o noktanın belirli bir fonksiyondan geçirilerek yeni nokta değerlerinin bulunmasına dayanır.

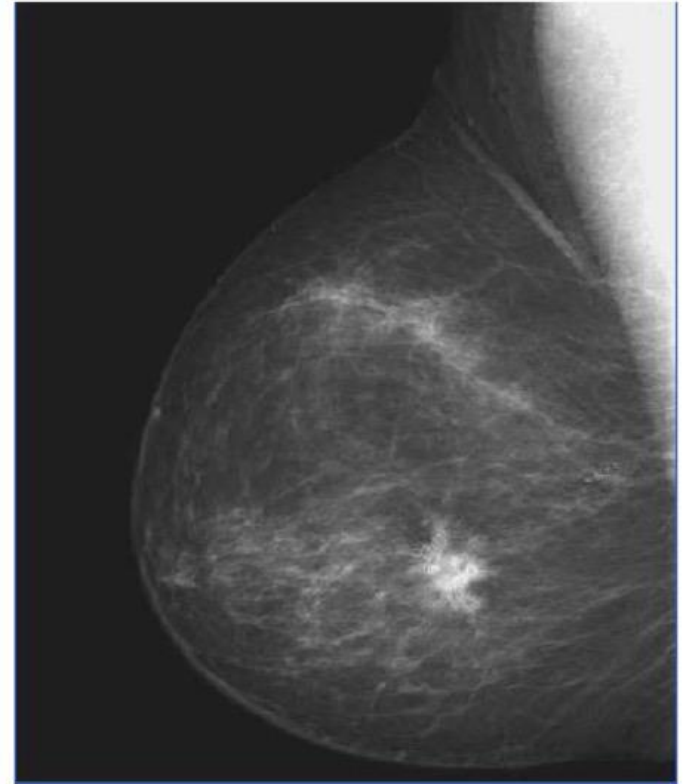


«imadjust» Fonksiyonu

**Bir radyolojik görüntü üzerine inceleme
(Sayısal Mamografi image'ı)**

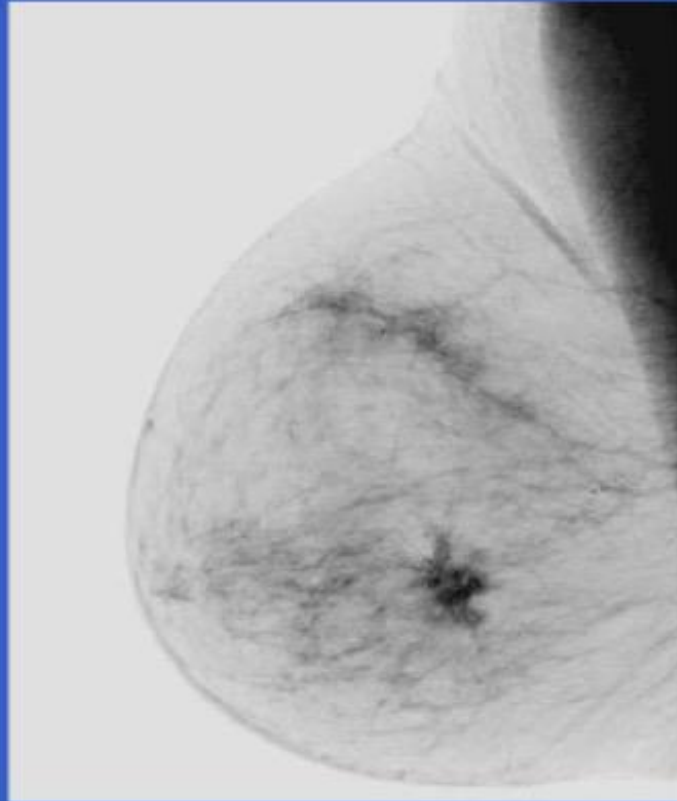
```
>> f=imread('breast.tif');
```

Resimde; göğüs üzerinde
küçük bir lezyon göze
çarpmaktadır.



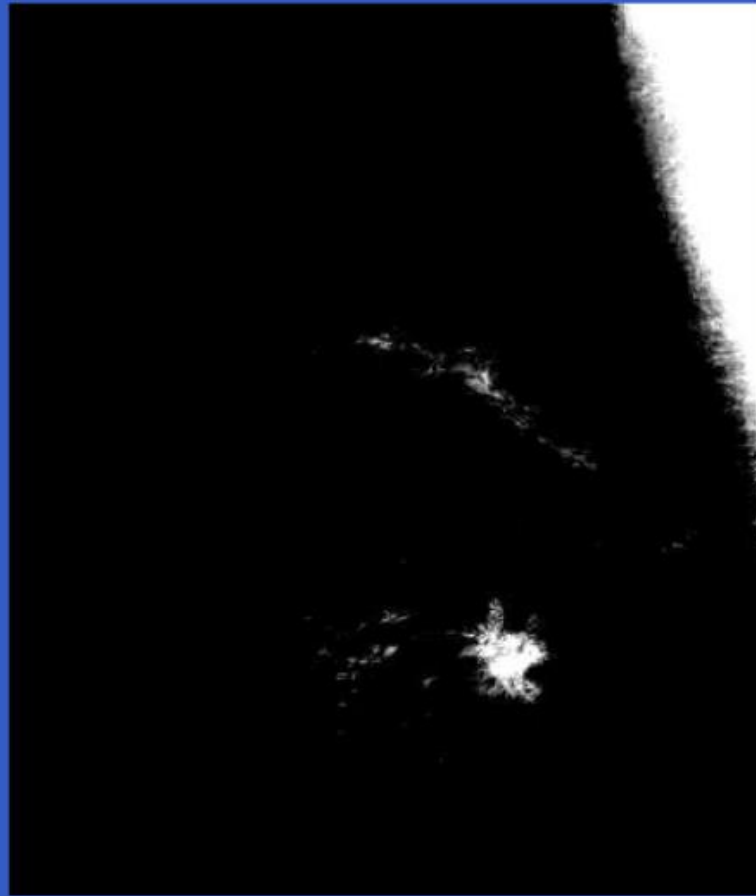
«imadjust» Fonksiyonu-Resmin Negatifi

```
>> g1=imadjust(f,[0 1],[1 0]);
```



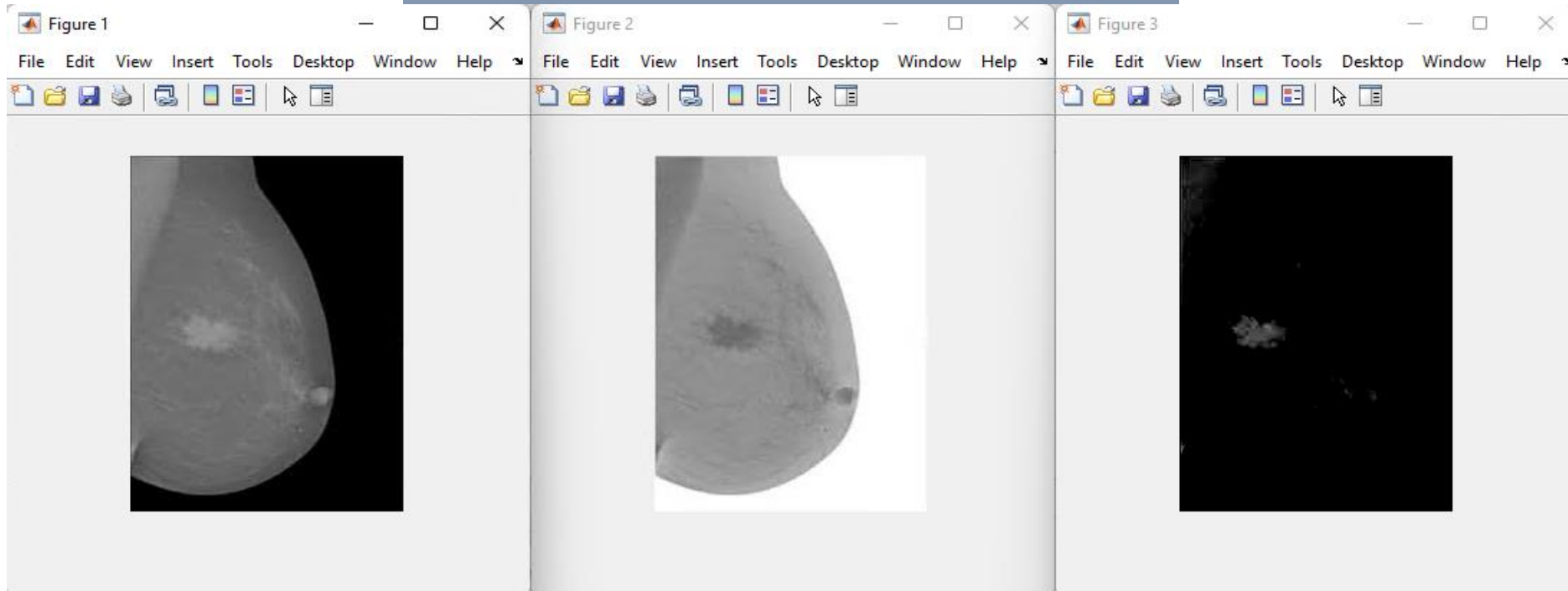
«imadjust» Fonksiyonu-Resmin Anlaşılması

```
>> g2=imadjust(f,[0.5 0.75],[0 1]);
```



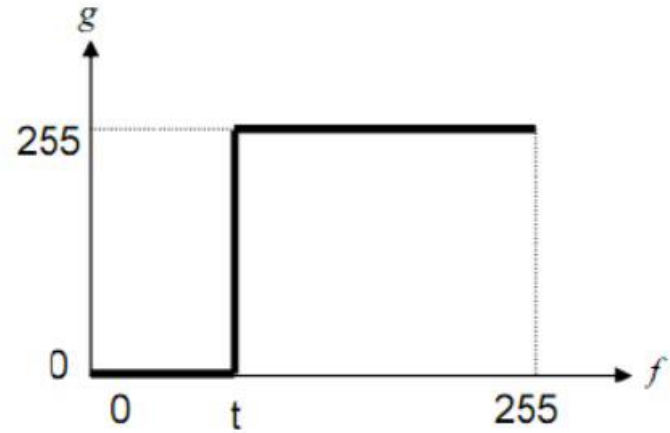
«imadjust» Fonksiyonu-Resmin Anlaşılması

```
clear all;clc;close all;  
f=imread('C:\Users\asus\Desktop\breast.jpg');  
f=rgb2gray(f);  
g1=imadjust(f,[0 1], [1 0]);  
g2=imadjust (f,[0.5 0.75], [0 1]);  
imshow(f), figure, imshow(g1); figure, imshow(g2);
```



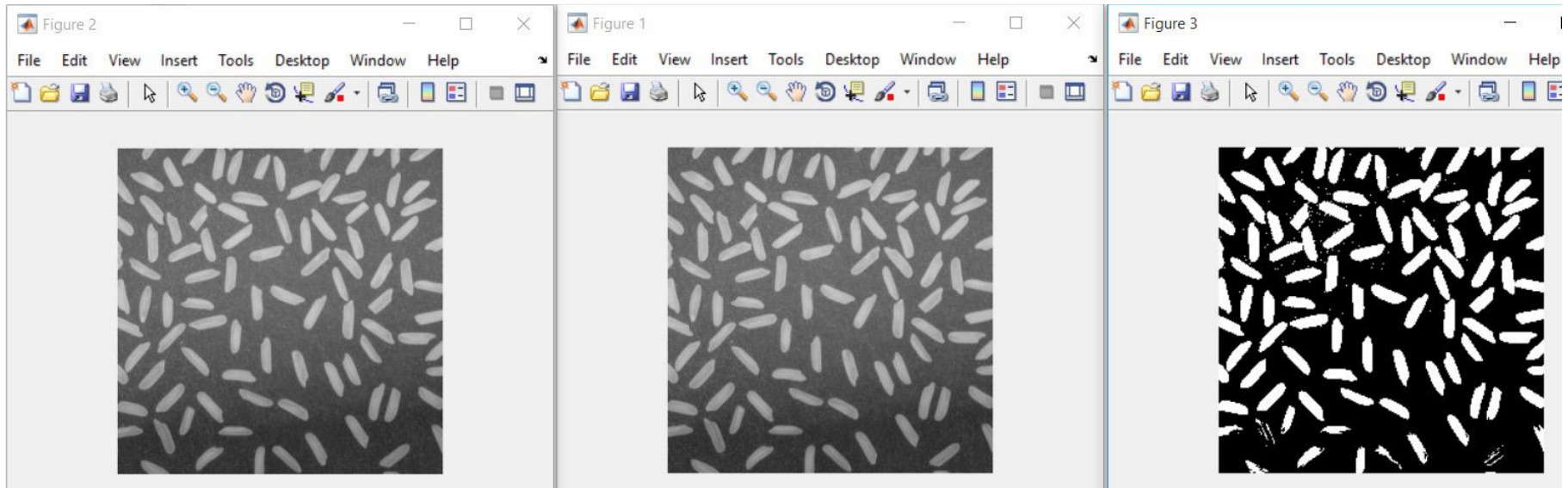
Eşikleme (Thresholding)

İmge ışıklılık değerlerinin belirli bir değerden büyüklük/küçük olmasına göre ikili (binary) bir imge oluşturulur. Bu ikili imgedeki renkler siyah ve beyazdır.



Eşikleme (Thresholding)

```
clear all; clc; close all;  
imgIkk = imread('rice.png');  
imge = im2double(imgIkk);  
figure;imshow(imgIkk);  
figure;imshow(imge);  
figure;imshow(imge>0.5);  
% Ekrana sadece 0.5 değerinden yüksek pikselleri (1) getirir, diğerlerini 0 a eşitler.
```

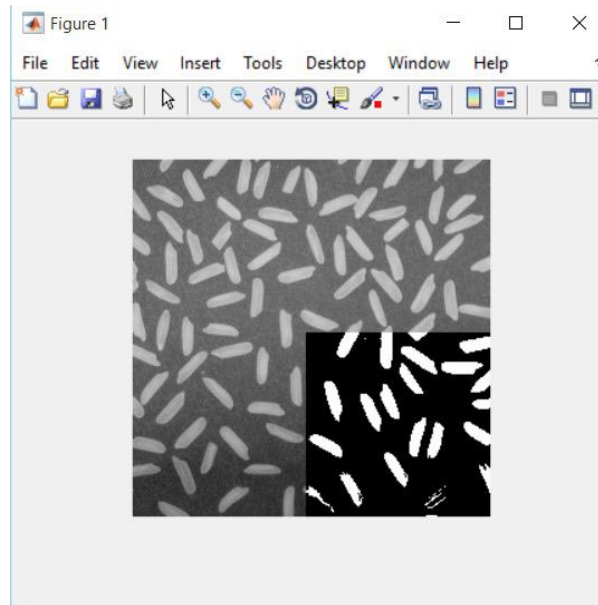


Kısmi Eşikleme (Thresholding)

```
clear all; clc; close all;
imgllk = imread('rice.png');
imge = im2double(imgllk);
imshow(imge>0.5) % Ekrana sadece 0.5 değerinden yüksek pikselleri getirir, diğerlerini 0 a eşitler.
imge3 = imge;
imge3(:,1:255)=imge(:,1:255)>=0.9;
imge3(:,200:255) = 1;
% Görüntüde kısmi eşikleme yapılmıştır, yani görüntü 255. piksele kadar eşiklenmiş oldu.
%Eşikleme > operatörü ile yapıldı, bu şu anlama geliyor, imge3 orjinal imgeden sadece 0.9
%üzerindeki değerleri alır, yoksa almaz ve o aralık 0 olur. Orjinal göründe beyaza yakın 0.9
%değeri olmadığı için bu değerin altındaki çoğunluk yerler (255 piksele kadar) sıfıra
%eşitlenmiş olur.
figure, imshow(imge3,[])
```

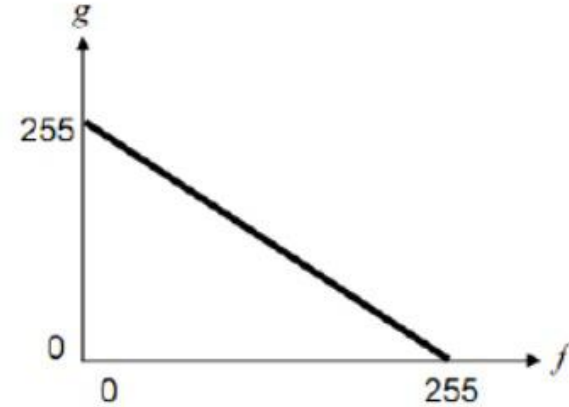
Kısmi Eşikleme (Thresholding)

```
clear all; clc; close all;  
imgllk = imread('rice.png');  
imge = im2double(imgllk);  
imge4 = imge;  
imge4(125:256,125:256)=imge(125:256,125:256)>=0.5;  
figure, imshow(imge4,[])
```



Piksel Tersleme (Negation)

İmge ışıklılık değerlerinin doğrusal biçimde ters çevrilmesi ise gerçekleştirilir.



Piksel Tersleme (Negation)

```
clear all; clc; close all;  
imgIkk = imread('cameraman.tif');  
imgNeg = 255-imgIkk;  
figure;  
subplot(1,2,1); imshow(imgIkk);  
title('Resmin orjinali');  
subplot(1,2,2); imshow(imgNeg);  
title('Resmin negatif');
```

256x256 uint8			
	1	2	3
1	156	159	158
2	160	154	157
3	156	159	158
4	160	154	157

256x256 uint8			
	1	2	3
1	99	96	97
2	95	101	98
3	99	96	97
4	95	101	98

Resmin orjinali



Resmin negatif





TEŞEKKÜRLER