

# SAYISAL GÖRÜNTÜ ÖRNEKLEME ve NİCELEME İKİLİ GÖRÜNTÜ İŞLEME

# Sayısal Görüntü

- İkili (Binary) Görüntü
- Gri Ölçekli (Grayscale) Görüntü
- Renkli (Colour) Görüntü

# Sayısal Görüntü

- Analog bir görüntü ve bu görüntünün örneklenmesi ile elde edilen sayısal görüntünün matrissel içeriği şöyle örneklendirilebilir.



*Gerçek resim*



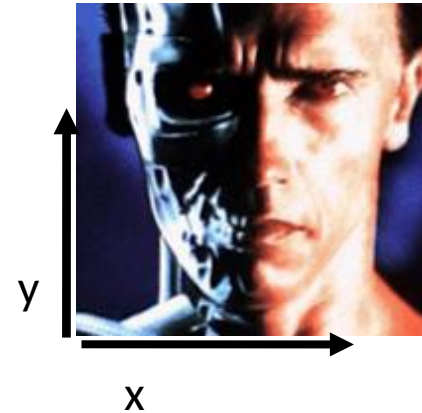
```
103 56 12 30 56 12 34 30 1 78 79 21 145 156 52 136 143 65 115 129 41 128 143 50 85
106 11 74 96 14 85 97 23 66 74 23 73 82 29 67 76 21 40 45 7 35 39 9 94 54 19
42 27 6 19 10 3 59 80 28 162 107 41 206 88 83 204 75 54 197 82 63 178 63 48 158 82
48 148 49 40 52 65 21 60 68 11 40 51 17 35 37 0 28 29 0 83 50 15 2 0 1 13 14
8 243 173 161 231 140 89 238 142 89 230 143 90 210 126 79 184 88 48 152 69 35 123 51
27 104 41 23 55 45 9 36 27 0 28 28 2 29 28 7 40 28 16 13 13 1 224 167 112 240
174 80 227 174 78 227 176 87 233 177 94 213 149 78 196 123 57 141 72 31 108 53 22 121
62 22 128 50 24 101 48 35 16 21 1 12 5 0 14 16 11 3 0 0 237 178 83 244 208 123
241 235 144 238 223 147 221 190 108 215 170 77 190 135 52 138 93 36 75 35 7 113 56 25
156 83 38 107 52 21 31 14 7 9 8 0 20 14 12 255 214 112 242 215 108 246 227 123 239
232 152 228 209 123 232 193 98 205 162 84 178 133 47 142 90 32 19 19 27 89 53 21 171
116 48 114 64 29 75 48 24 10 9 5 11 16 9 237 180 62 249 221 122 141 225 129 240 219
126 240 199 93 218 173 69 188 135 33 218 166 79 189 184 93 136 154 65 112 68 37 191 153
80 122 74 26 80 51 19 19 37 47 18 37 32 223 177 83 235 206 105 243 218 125 238 208
103 221 188 83 228 204 98 224 220 123 210 194 159 192 159 62 150 96 40 116 73 28 146 104
48 109 69 24 75 48 18 27 33 33 47 100 118 216 177 96 223 189 91 239 209 111 236 213
117 217 200 168 218 200 100 218 206 104 207 175 76 177 131 54 142 88 41 106 65 22 103
59 22 93 53 18 76 50 17 9 10 2 54 76 74 108 111 102 218 194 108 228 203 102 228 200
100 212 180 79 226 182 85 198 156 62 186 138 54 155 106 37 132 62 33 65 51 14 67 48
15 81 46 14 16 15 0 11 6 0 64 90 91 54 90 90 220 185 97 212 190 105 214 177 86 208
185 71 196 150 64 175 127 42 170 117 49 139 89 30 102 53 12 84 43 13 79 46 15 72 42
14 10 13 4 12 8 0 89 104 110 58 96 109 130 128 115 196 154 82 198 148 68 163 136 70
174 125 98 169 120 54 146 97 41 118 67 24 90 52 16 75 46 16 58 42 19 13 7 9 10 5
0 16 11 3 66 111 116 70 100 102 78 103 98 57 71 82 182 111 98 141 96 37 152 102 51
130 86 31 110 63 21 83 44 11 69 42 12 28 8 0 7 5 10 16 4 0 17 10 2 30 20 10
58 88 96 53 88 94 59 91 102 69 99 110 54 80 79 23 69 85 31 34 25 53 41 25 21 2
0 8 0 0 17 10 4 11 0 0 34 21 13 47 35 23 38 26 14 47 35 23
```

*Sayısal resim*

# Sayısal Görüntü

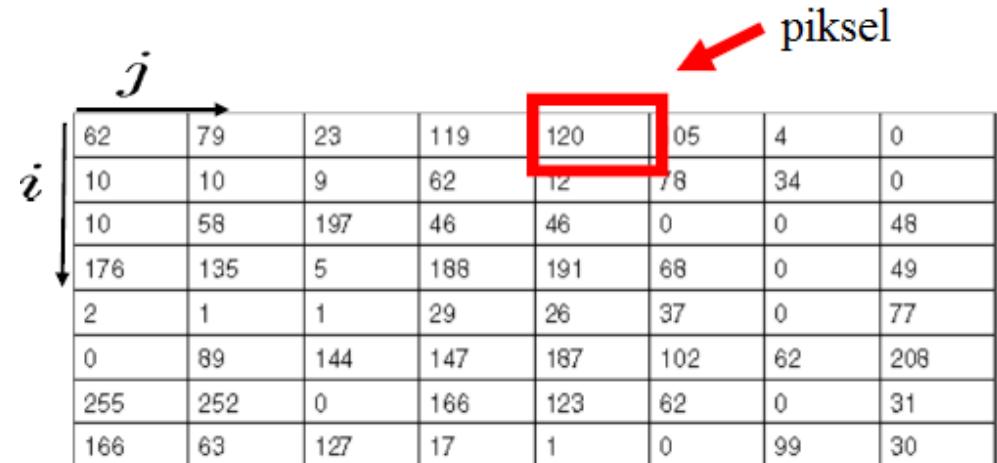
- Siyah-beyaz bir görüntü, iki boyutlu bir fonksiyon olarak  $f(x,y)$  şeklinde ifade edilir.
- Renkli bir görüntü, iki boyutlu bir vektör fonksiyon olarak aşağıda şekilde ifade edilir.

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$



# Sayısal Görüntü

- Sayısal siyah-beyaz bir görüntü, fonsiyonun tanımlı olduğu bölge üzerinde sonlu sayıda nokta alınarak (**örnekleme**) ve fonsiyonun alabileceği değerler sonlu sayıda tamsayı ile sınırlandırılarak (**niceleme**) oluşturulur.
- Dolayısıyla, sayısal siyah-beyaz görüntüler tamsayı değerlerden oluşan bir matris olarak ifade edilir. Matrisin herhangi bir elemanına bir PİKSEL denir.
- $[i,j]$  koordinat bilgisini gösterirken,  $I[i,j]$  değeri bu koordinata karşılık gelen pikselin yoğunluğunu gösterir. Tek renkli yoğunluğu belirtmek için gri düzeyi(gray level) terimi kullanılır.



	$j$								
		62	79	23	119	120	05	4	0
$i$		10	10	9	62	12	78	34	0
		10	58	197	46	46	0	0	48
		176	135	5	188	191	68	0	49
		2	1	1	29	26	37	0	77
		0	89	144	147	187	102	62	208
		255	252	0	166	123	62	0	31
		166	63	127	17	1	0	99	30

# Sayısal Görüntü

- Görüntünün sayısallaştırılması aşamasında iki boyutlu dizinin boylarına ve her bir piksel için kaç farklı gri düzeyine izin verileceğine karar verilmelidir.
- Bu değerler genelde 2'nin kuvveti olan tamsayılardan seçilir.
  - M ve N görüntü dizisinin boyları ve k da bir pikselin yoğunluğunu göstermek için kullanılan bit sayısı olmak üzere;

$$N=2^n \quad M=2^m \quad \text{and} \quad G=2^k$$

 G = gri düzey sayısı

- Her bir düzey bir gri ölçeğine (gray scale) sahiptir ve düzeyler arasındaki geçiş düzenli bir şekilde dağılır (L adet gri ölçeği).
- Örneğin, izin verilen maksimum gri düzeyi 256 olarak seçildiğinde, 0 en koyu rengi (siyah) ve 255 en parlak rengi(beyaz) gösterir.

# Sayısal Görüntü



n=2



n=4



n=8



n=16



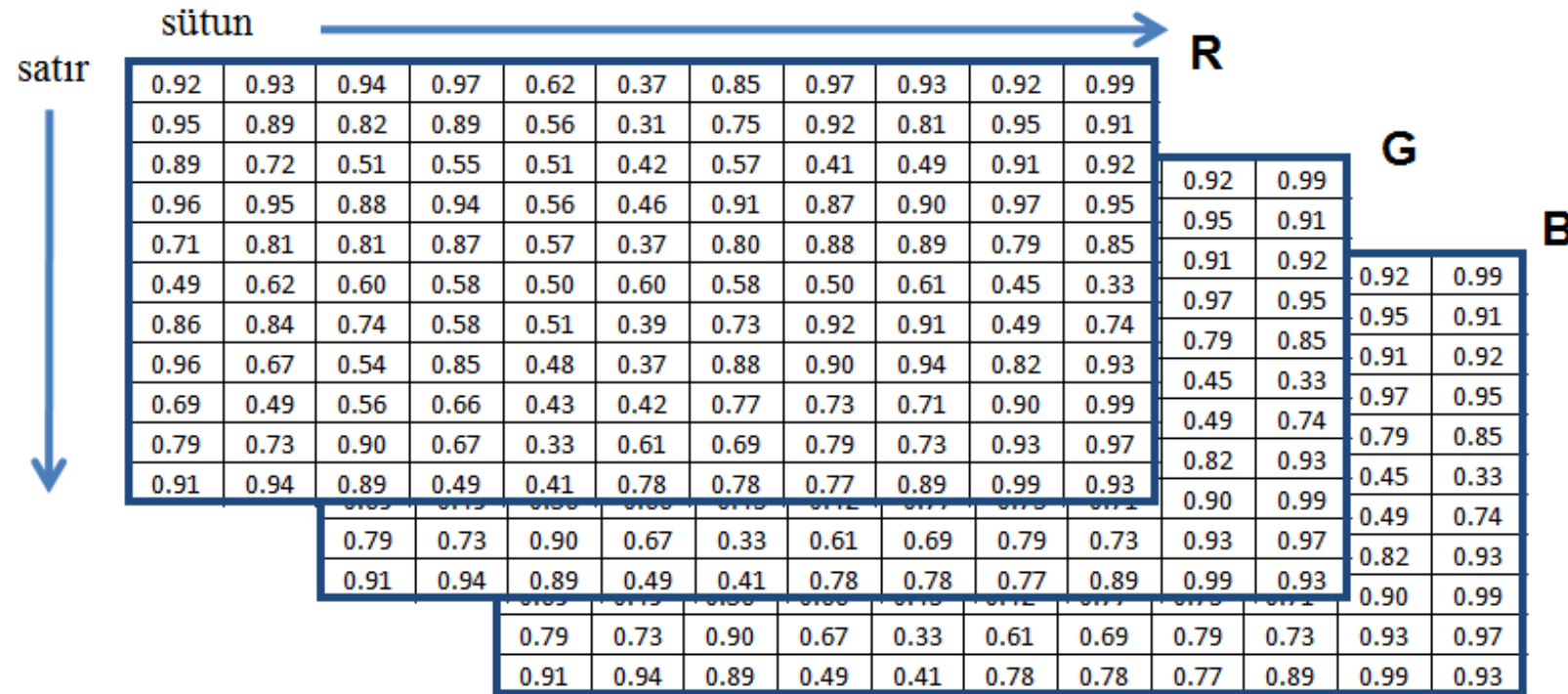
n=64



n=256

# Sayısal Görüntü

- Sayısal renkli bir görüntü, üç matris olarak ifade edilir.

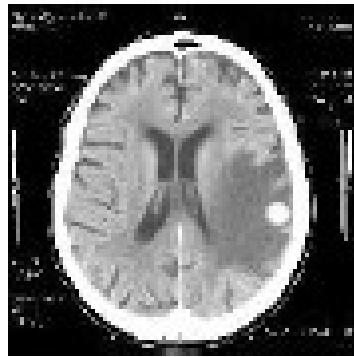




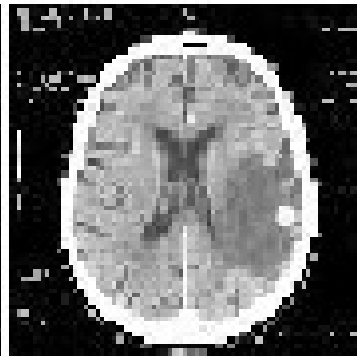
# Uzaysal Çözünürlük

- Bir resmin uzaysal çözünürlüğü (Spatial Resolution), o resmin 1 pikselinin fiziksel büyüklüğüne eşittir.
- Kısaca, bir resmin detaylanabilir en küçük parçasıdır.
- Ayrıntıların fark edilebildiği en küçük çözünürlüktür.

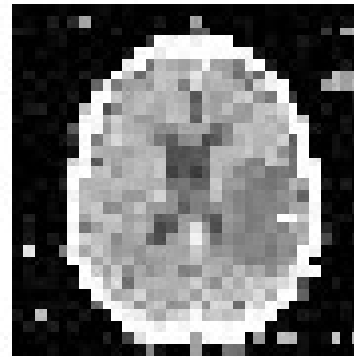
# Uzaysal Çözünürlük



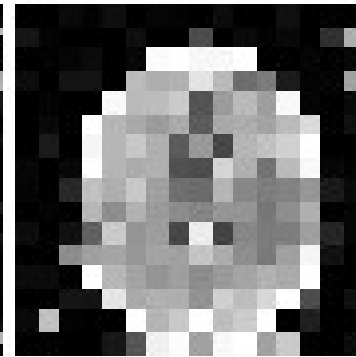
128x128



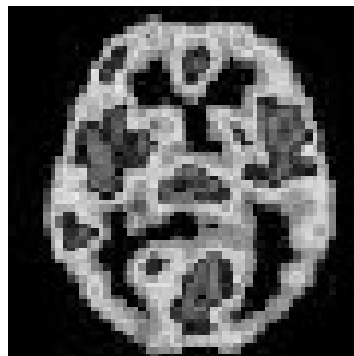
64x64



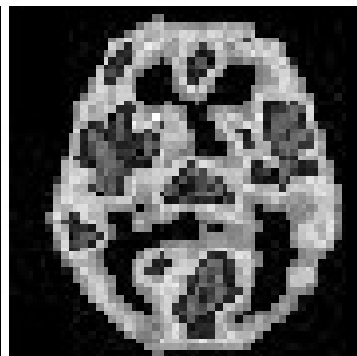
32x32



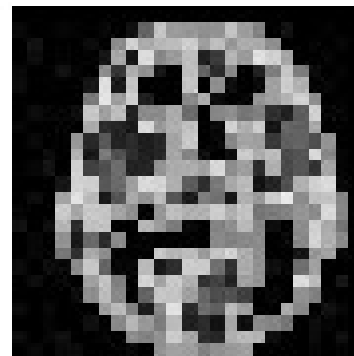
16x16



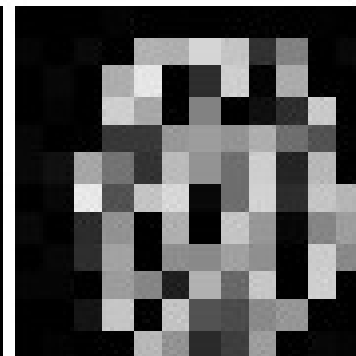
100x100



50x50

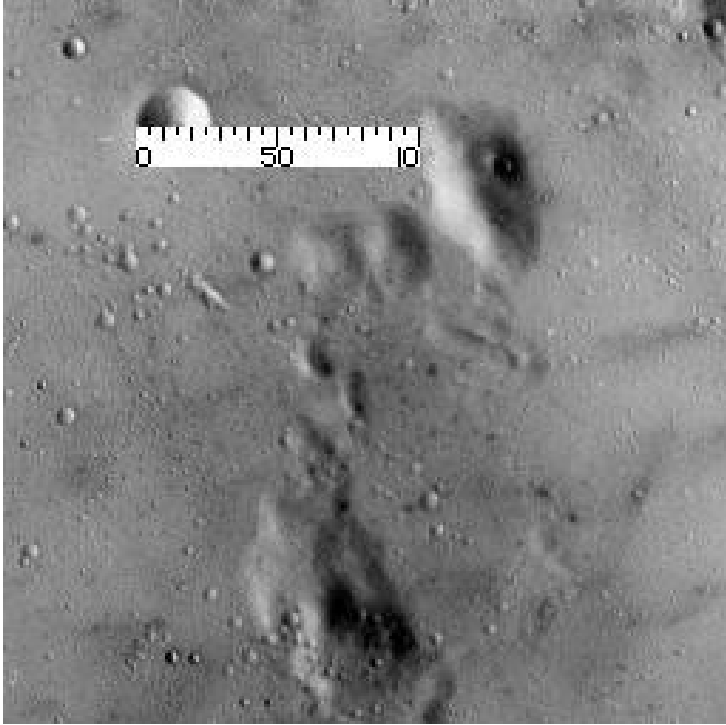


24x24

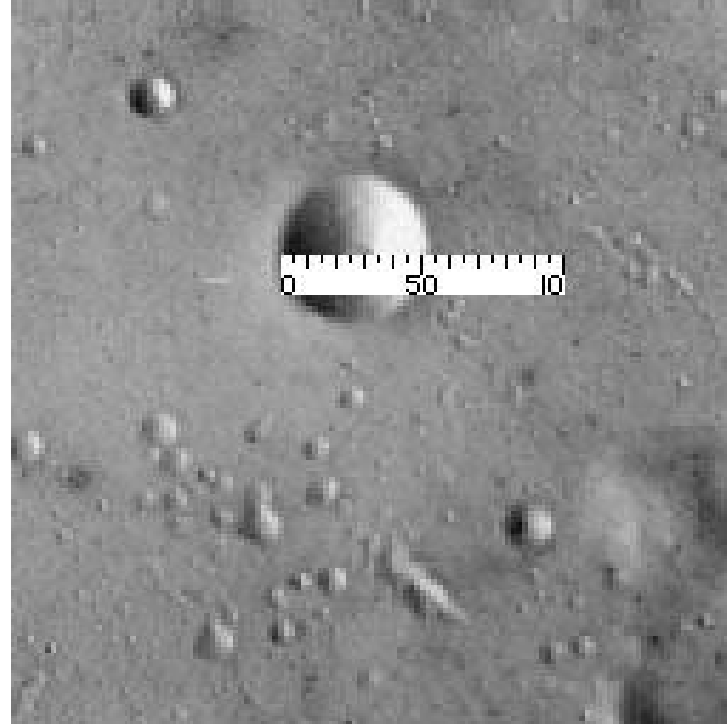


12x12

# Uzaysal Çözünürlük



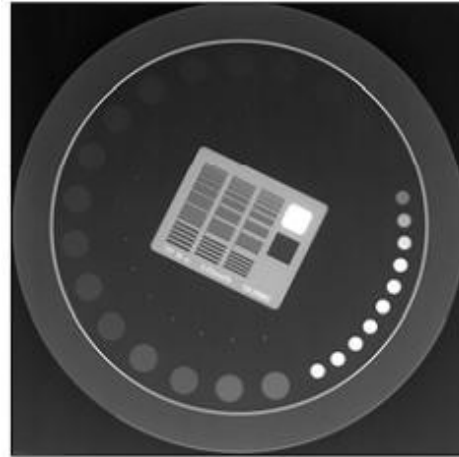
(a) 20km/piksel



(b) 10km/piksel

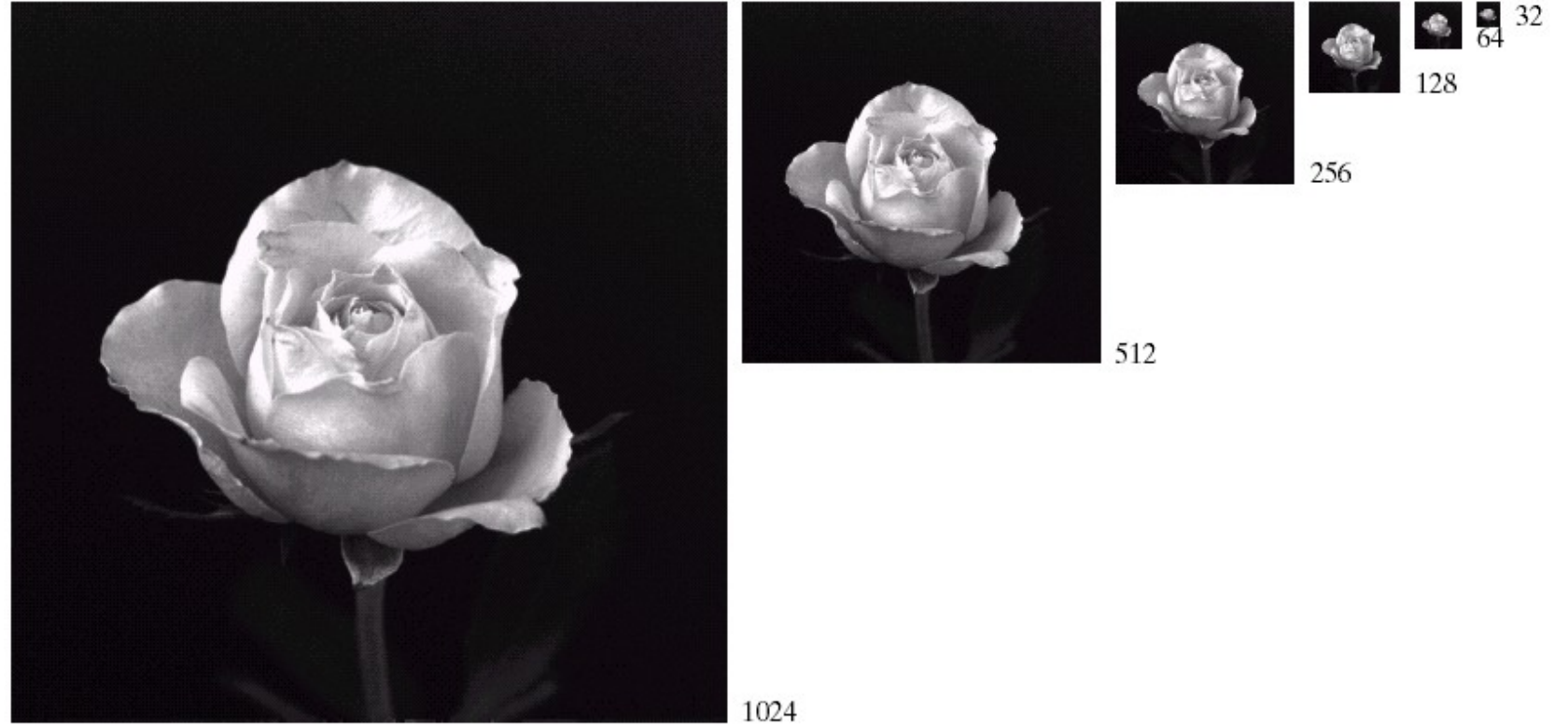
# Gri Düzey Çözünürlüğü (Gray Level Resolution)

- Ayrıntıların fark edilebildiği en küçük gri düzey sayısıdır.
- Gri-düzey sayısı L genelde 2'nin kuvvetidir.
- Düzey sayısına Analog -> Sayısal çevirici tarafından karar verilir.



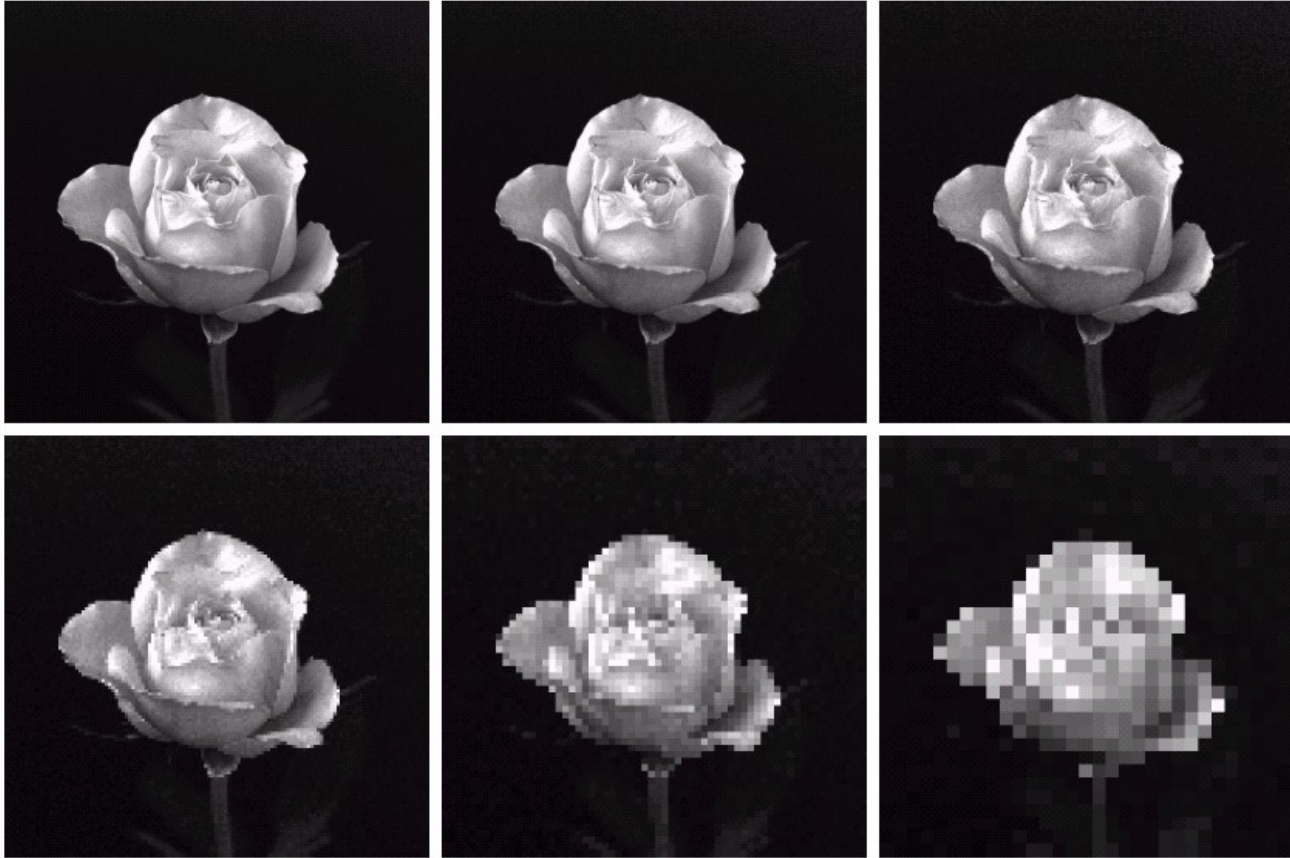
# Uzaysal Çözünürlük – Gri Düzey Çözünürlüğü Örnekler

- Gri düzey sayısı sabitken, uzaysal çözünürlük azaltılmış.
- $1024 \times 1024$  görüntüden  $32 \times 32$  ye kadar alt örnekleme yapılmış.
- 8-bit görüntü –256 gri düzeyi.



**FIGURE 2.19** A  $1024 \times 1024$ , 8-bit image subsampled down to size  $32 \times 32$  pixels. The number of allowable gray levels was kept at 256.

# Uzaysal Çözünürlük – Gri Düzey Çözünürlüğü Örnekler



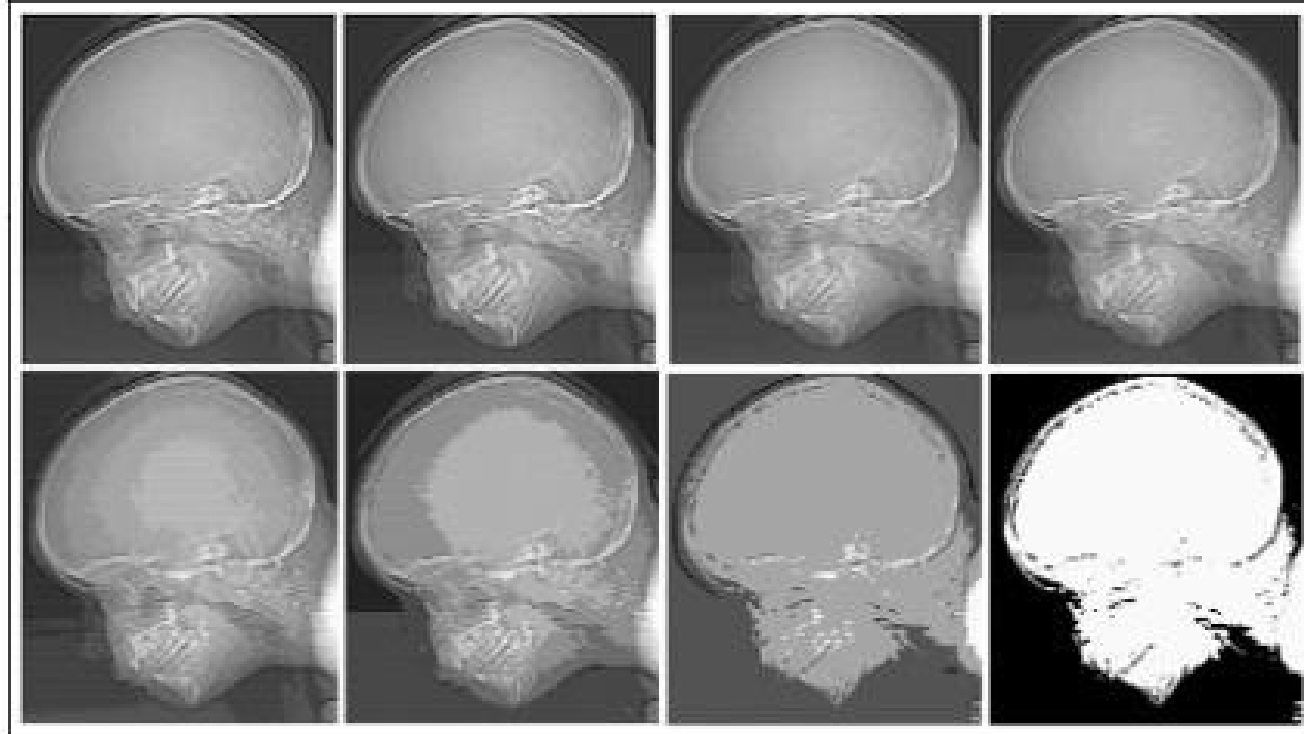
a b c  
d e f

**FIGURE 2.20** (a)  $1024 \times 1024$ , 8-bit image. (b)  $512 \times 512$  image resampled into  $1024 \times 1024$  pixels by row and column duplication. (c) through (f)  $256 \times 256$ ,  $128 \times 128$ ,  $64 \times 64$ , and  $32 \times 32$  images resampled into  $1024 \times 1024$  pixels.

- Gri düzey sayısı sabitken, uzaysal çözünürlük arttırılmış.
- $32 \times 32, 64 \times 64, \dots, 512 \times 512$  görüntüler  $1024 \times 1024$  çözünürlüğe sahip olacak şekilde yeniden örneklenmiş.

# Uzaysal Çözünürlük – Gri Düzey Çözünürlüğü Örnekler

- Bu örnekte, uzaysal çözünürlük sabit tutulurken gri-düzey çözünürlüğü azaltılmış.
- Sol üst görüntüde 8-bit (256 düzey) kullanılırken en sağ-alt görüntüde sadece 1-bit (2 düzey) kullanılmış.



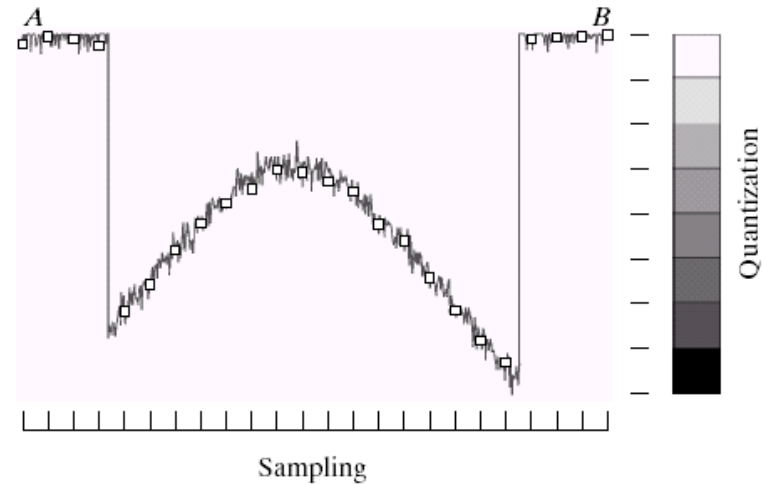
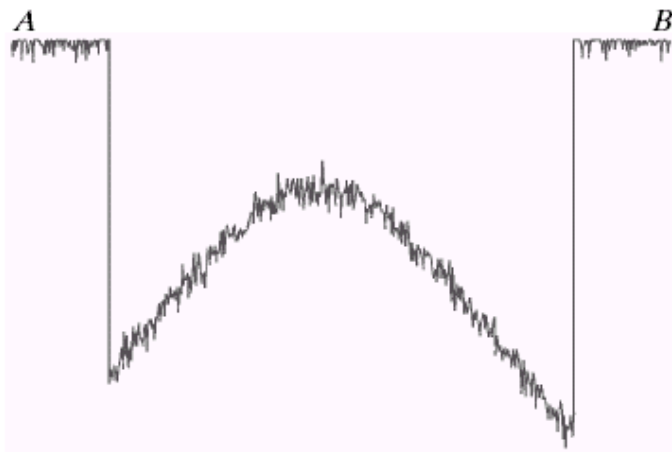
# Örnekleme ve Niceleme

- Bir görüntü  $x, y$  koordinatlarında ve genlik boyutunda sürekli.
- Bu görüntüyü sayısal forma dönüştürmek için her iki koordinatta ve genlikte örneklemeliyiz.
- Bu işlem ile koordinatların sayısallaştırılmasına **örnekleme** denir.
- Genlik değerlerinin sayısallaştırılmasına ise **niceleme** denir.



# Örnekleme ve Niceleme

- Aşağıdaki soldaki şekilde sürekli bir görüntünün genlik değerlerini (gri seviye) gösteren tek boyutlu bir fonksiyon görülmektedir.
- Rastgele değişiklikler görüntüdeki gürültüden dolayıdır.
- Bu fonksiyonu örnekleme için A-B arasında eşit uzunlukta bölünmüş bir eksen alınır (sağdaki şekil).



# Örnekleme ve Niceleme

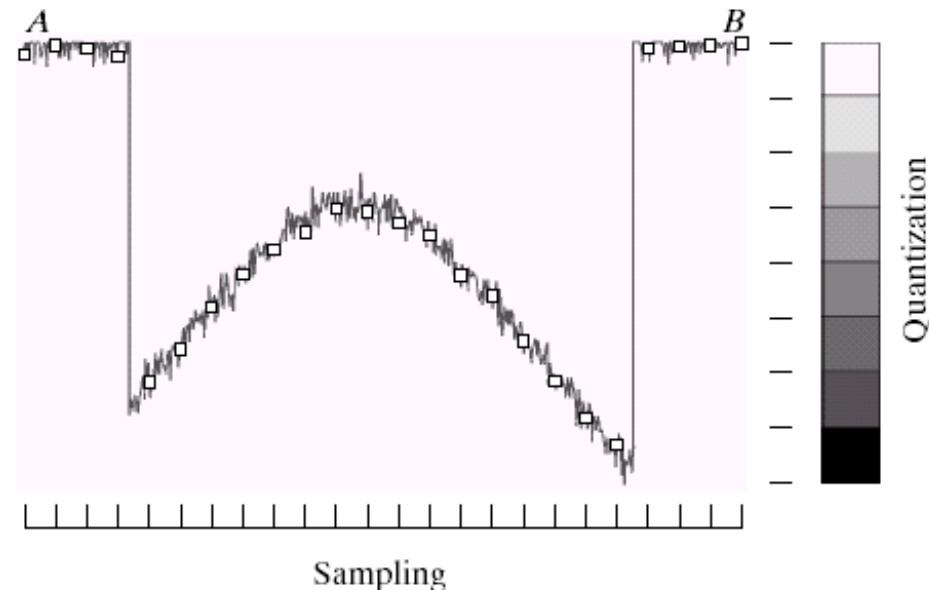
- Orijinal görüntü kalitesine optimum düzeyde yaklaşmak için
  - kaç tane gri-düzeyi kullanılmalı?
  - Ne kadar(hangi sıklıkta) örneklem alınmalı?
- Bir görüntünün kalitesi kullanılan piksel ve gri-düzey sayısına göre belirlenir.
- Bu parametrelerin değerleri arttıkça sayısal görüntünün kalitesi orijinal görüntüye bir o kadar yaklaşır.
- Ama,  $M, N$  ve  $k$  değerleri arttıkça görüntüyü saklamak ve elde etmek için gerekenen ihtiyaçlarda doğru orantılı olarak artar.

# Örnekleme ve Niceleme

- Her bir örnekleme alanını belirlemek için yatay eksene denk gelen fonksiyon üzerindeki her bir noktada işaretleme yapılır.
- Örnekleme noktaları fonksiyonun üzerine koyulan küçük beyaz karelerle gösterilmiştir.
- Bu küçük karelerin hepsi örneklenmiş fonksiyonu verir.
- Bununla birlikte örneklerin değerleri sürekli gri seviye değerlerindedir.
- Sayısal fonksiyon formu için gri seviyeli değerler ayırık miktarlara dönüştürülmelidir (niceleme).

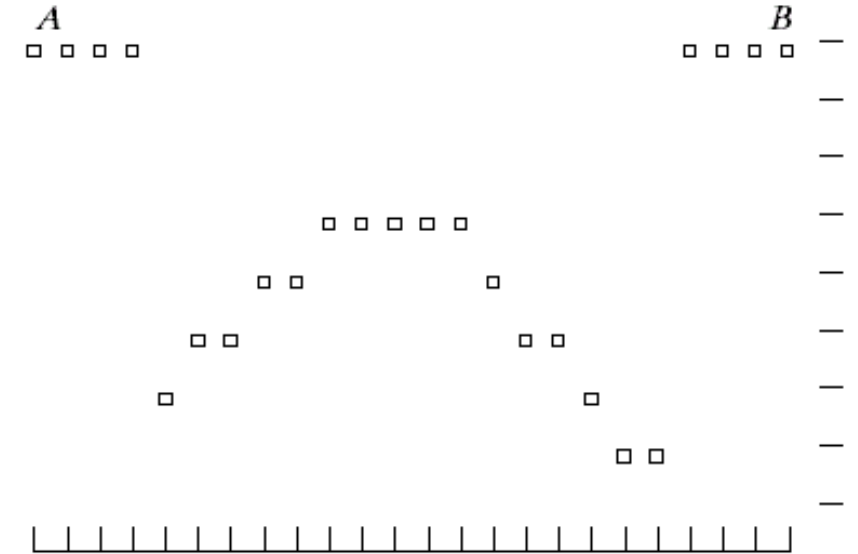
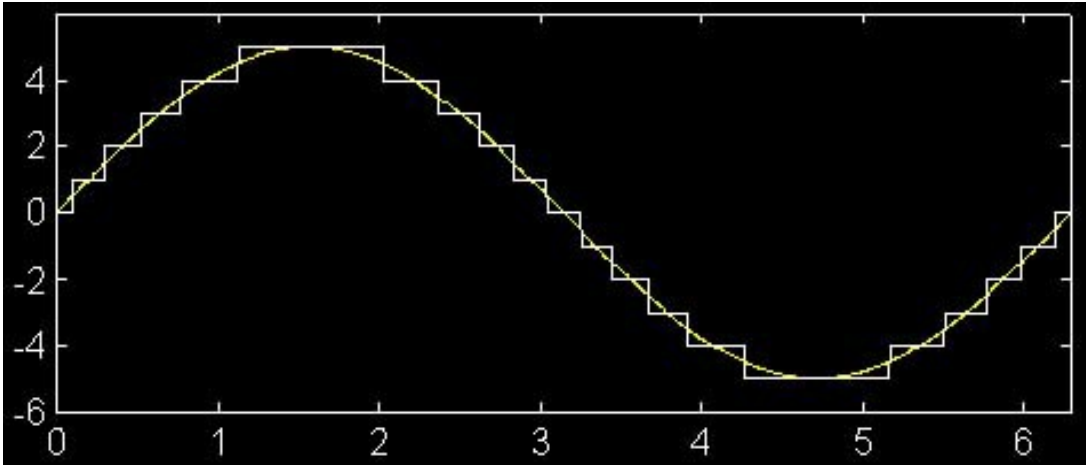
# Örnekleme ve Nicelleme

- Aşağıdaki şekilde sağ taraf siyah – beyaz aralığında sekiz ayrık gri seviyeye bölünmüştür.
- Dikeydeki işaretler bu her bir sekiz gri seviyeyi temsil etmektedir.
- Bu sürekli görüntüdeki gri seviyeler sağ taraftaki sekiz ayrık gri seviyesine göre nicelenir.



# Örnekleme ve Niceleme

- Bu niceleme işlemi dikeydeki beyaz karelerin yaklaşık olarak nerde olduklarına bağlıdır.
- Sayısal örnekleme sonuçları (örnekleme ve niceleme) aşağıdaki şekilde görülmektedir.
- Bu işlem, görüntünün en üstünden başlayıp her bir satır için gerçekleştirildiğinde 2 boyutlu bir görüntü elde edilir.



# Örnekleme ve Niceleme

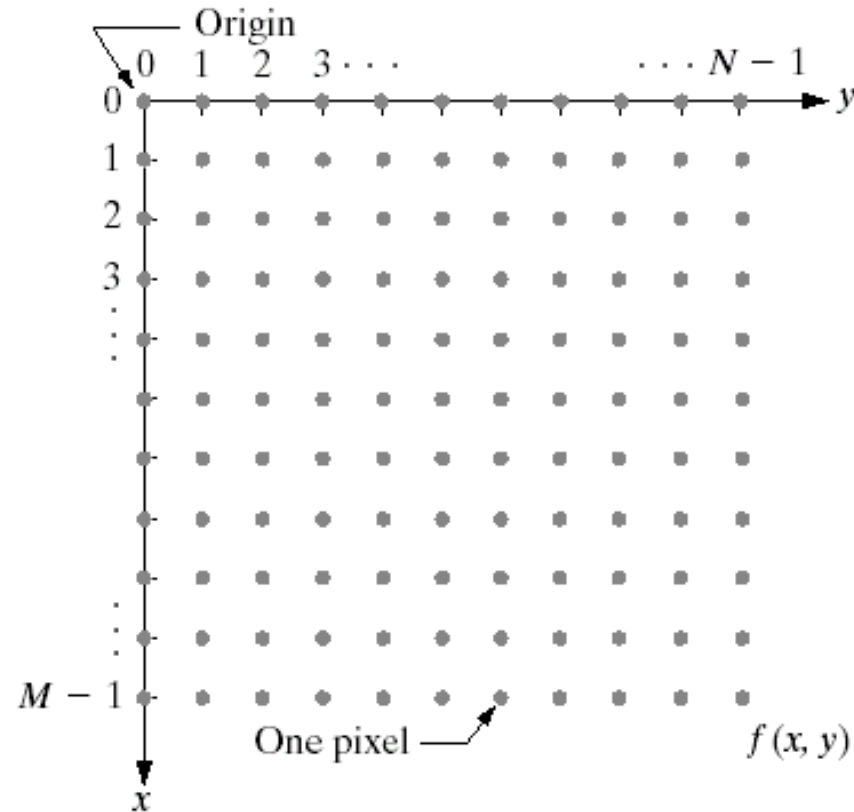
- Örnekleme ve nicelemenin sonucu gerçek sayılardan oluşan bir matristir.
- $f(x,y)$  görüntüsü örneklendiğinde sonuç olarak sayısal görüntü M satır ve N sütundan oluşacaktır.
- $(x,y)$  koordinatlarının değerleri ayrık niceelenmiştir.
- İfade etmenin kolay olması için bu ayrık koordinatlar için tamsayı değerleri kullanılır.
- Böylece herhangi bir koordinatın değeri  $(x, y) = (0, 0)$  olarak gösterilir.

# Örnekleme ve Niceleme

- Aynı satırdaki bir diğer koordinat ise  $(x, y) = (0,1)$  olarak gösterilir.
- $(0,1)$  ifadesi ilk satır boyunca yapılan 2. örnekleme göstermektedir.
- Bir görüntü örneklendiğinde bu değerler fiziksel koordinatlardaki gerçek değerleri ifade etmemektedir.

# Örnekleme ve Nisceleme

- Aşağıda verilen koordinat genel olarak kullanılan koordinat eksenidir.





# Örnekleme ve Niceleme

- Yukarda belirtilen notasyon (sayısal ifade) MxN boyutundaki sayısal bir görüntünün aşağıdaki gibi yazılabilmesine imkan verir:

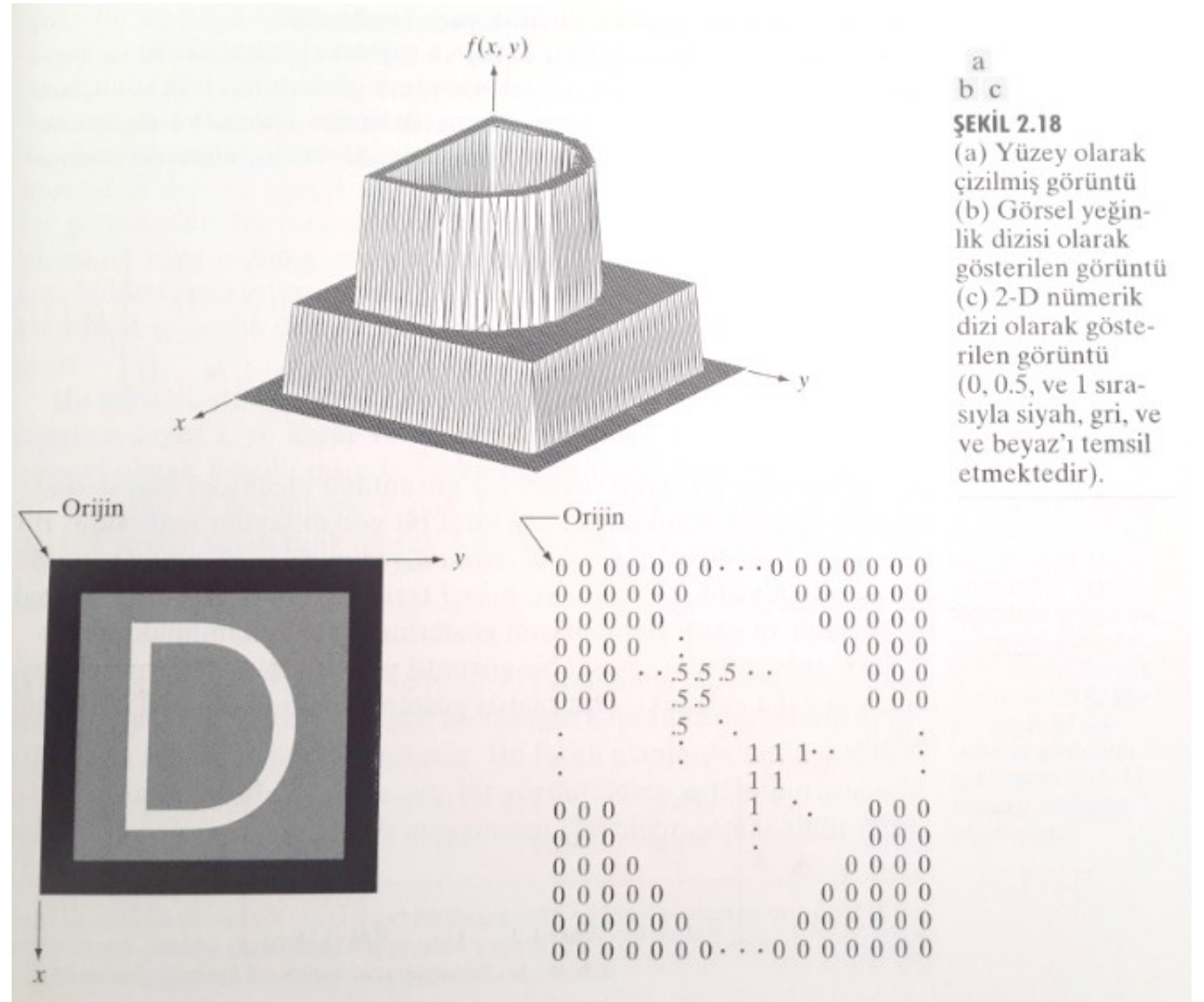
$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix}$$

# Örnekleme ve Niceleme

- Bu denklemin sağ tarafı sayısal görüntüyü tanımlar.
- Matris dizisindeki her bir eleman görüntünün her bir elemanına yani piksele karşılık gelir.
- Bazı görüşlere göre sayısal görüntü ve elemanlarının geleneksel matris ifadesi ile gösterilmesi daha avantajlıdır:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

# Örnekleme ve Niceleme

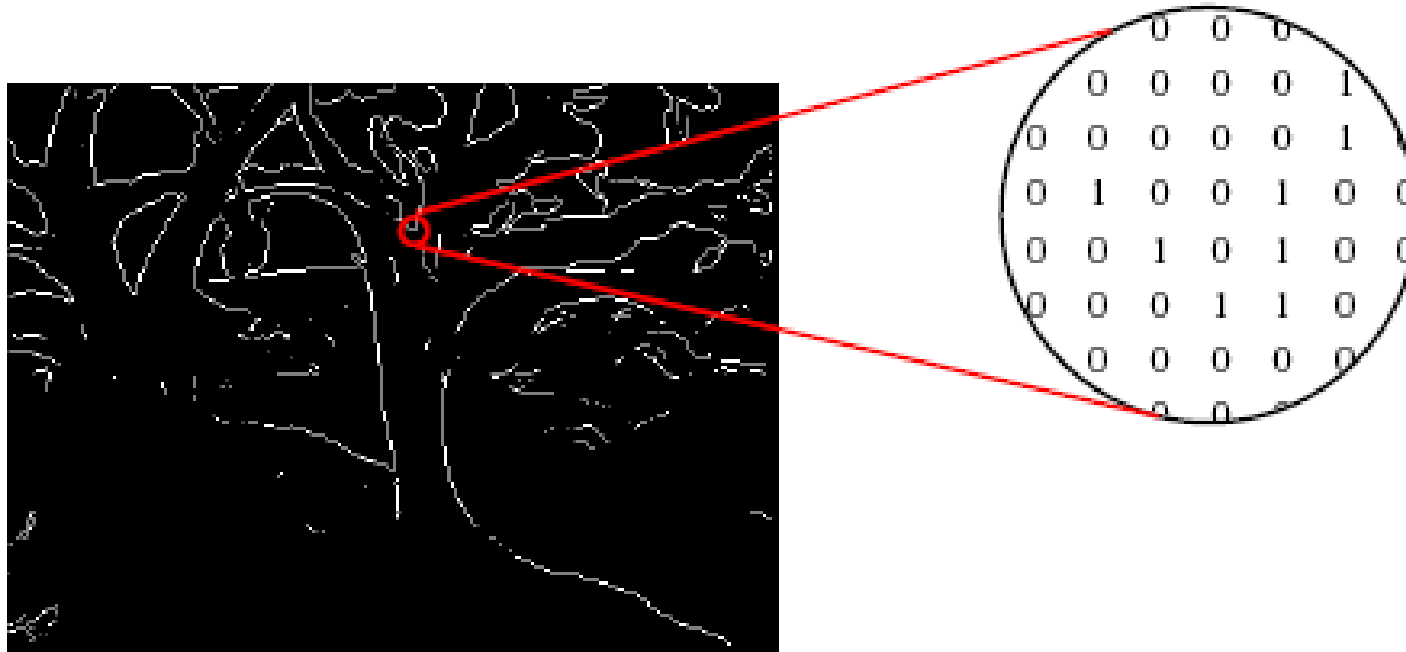


# İkili Görüntü

- İkili bir görüntü sadece 0 ve 1 değerlerinden oluşur ve iki boyutlu matrisler halinde saklanır.
- İkili görüntü, gri seviyeli bir görüntünün ön plandaki piksellerinin alındığı bir işlem sonucunda oluşur. Geri kalan tüm pikseller arka plan pikselleri olarak adlandırılır.
- 1 değeri ön plan piksellerini ifade ederken, 0 değeri de arka plan pikselleridir.

# İkili Görüntü

This figure shows an example of a binary image.



# Maskeler

- Görüntü işleme uygulamalarında kullanılan en temel işlemdir.
- Convolution işlemi olarak bilinir. Maskenin piksel piksel resim üzerinde gezdirilmesidir.
- Maske üzerinde bulunan her bir değere ağırlık adı verilir.

# Maske Örnekleri

0	-1	0
-1	4	-1
0	-1	0

Laplacian

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

Sobel

1
1
1
1

# Maskeler

- Maske uygulaması sonucunda orijinal görüntü ile aynı boyutlu sonuç görüntüsü elde edilir.
- Maske her bir piksel üzerine yerleştirilir, her bir piksel maske ağırlıkları ile çarpılır ve toplanır.
- Bu değer çıkış görüntüsünde aynı piksel pozisyonuna yerleştirilir.



# Maskeler

40	40	80	80	80
40	40	80	80	80
40	40	80	80	80
40	40	80	80	80
40	40	80	80	80



1	2	1
2	4	2
1	2	1

a) original gray-tone image

b)  $3 \times 3$  mask

640	800	1120	1280	1280
640	800	1120	1280	1280
640	800	1120	1280	1280
640	800	1120	1280	1280
640	800	1120	1280	1280

c) result of applying the mask to the image

40	50	70	80	80
40	50	70	80	80
40	50	70	80	80
40	50	70	80	80
40	50	70	80	80



d) normalized result after division by the sum of the weights in the mask (16)

# Maskeler



$$\begin{matrix} * & \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} & = \end{matrix}$$



```
A=imread('tulips.bmp');  
B=[-1 0 1;-2 0 2;-1 0 1];  
A=rgb2gray(A);  
C=conv2(A,B,'same');  
imshow(C)  
imshow(A)
```

# Matematiksel Morfoloji

- Matematiksel morfoloji, geometrik objelerle ilgili işlem ve analiz yapma teori ve tekniğidir. Temeli küme teoremi, topoloji, rasgele fonksiyonlar ve kafes kuramına bağlıdır. En yaygın kullanımı sayısal görüntüler üzerindedir. Ancak bilindiği üzere grafikleri, yüzey ağları, katı cisimler ve pek çok uzaysal cisimlerde kullanılmaktadır.

# Kullanım Alanları

- Görüntü geliştirme
- Görüntü segmentasyonu
- Görüntü onarma
- Kenar yakalama
- Doku analizi
- Parçacık Analizi
- Genelleştirme
- İskelet belirleme
- Şekil analizi
- Görüntü Sıkıştırma
- Bileşen analizi
- Eğri keskinleştirme
- İnceltme
- Özellik ayırma
- Gürültü azaltma
- Boşluk azaltma

# Matematiksel Morfoloji

- İkili görüntü  $B$  ve yapısal eleman  $S$  gereklidir.
  - $S$  genelde küçük bir ikili görüntüdür.
  - $S$  herhangi bir formda olabilir.
  - $S$  in bir pikseli merkez olarak bulunur.
  - $S$  aynı maske gibidir.
- Merkez piksel görüntü üzerinde herhangi bir noktaya konarak istenilen işlem gerçekleştirilir.

# Yapısal Elemanlar

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Box(3,5)

	1	1	1	
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
	1	1	1	

Disk(5)

	1	1	1	
1				1
1				1
1				1
	1	1	1	

Ring(5)

1	1		
1	1		
1	1	1	1
1	1	1	1

1	1	1	1	1	1
1		1	1		1
1		1	1		1
1		1	1		1

1
1
1
1

Boşluklar "0" ı temsil etmektedir

# Temel Morfolojiler

- **Translation: Öteleme**

- $X_t = \{x + t \mid x \in X\}$

- **Dilation (Genişletme):** İkili görüntülerdeki nesneleri büyütmek için veya belirginleştirmek için kullanılır.  $B \oplus S$

S yapısal elemanı B görüntüsünün üzerine yerleştirildiği zaman S nin merkezi B görüntüsündeki herhangi bir piksele denk geliyorsa, S yapısal elemanının bütün elemanları B görüntüsüne yerleştirilir.

# Temel Morfolojiler

## İkili Görüntü B

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

# Yapısal Eleman-S

1	1	1
1	<b>1</b>	1
1	1	1

## Dilation Sonrası: $B \oplus S$

[illegible]



# Temel Morfolojiler

[illegible]
$$\begin{matrix} & & & & 1 \\ & & & 1 & \\ & & 1 & & \\ & 1 & & & \\ 1 & & & & \end{matrix}$$

# Temel Morfolojiler

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Şekil 1

**Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.**

Şekil 2

1  
1 1 1  
1

Şekil 1’de görülen orijinal resim yukarda verilen yapısal eleman kullanılarak genişletildiğinde Şekil 2’deki görüntü elde edilir.

# Temel Morfolojiler

- **Erosion (Aşındırma):** Kenar bölgelerden piksel yok eder.

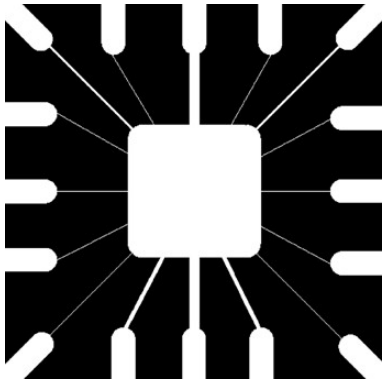
$$B \ominus S$$

S yapısal elemanı B görüntüsü üzerine yerleştirilir. S elemanı B görüntüsünün üzerine tam yerleştiğinde, yani hiçbir elemanı B görüntüsünün dışında kalmadığı takdirde, S elemanının merkezine denk gelen B görüntüsünün pikseli değiştirilmez. Eğer S yapısal elemanına ait herhangi bir eleman B görüntüsünün dışında kalıyorsa, S yapısal elemanının merkezine denk gelen B görüntüsündeki piksel silinir.

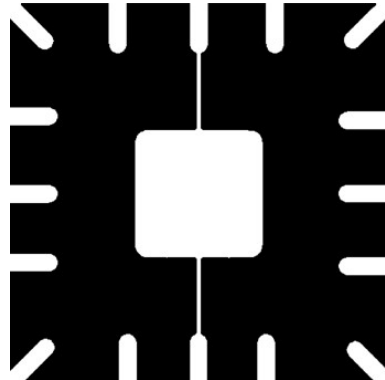
# Temel Morfolojiler

[illegible]
$$\begin{array}{ccc} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{array}$$
[illegible]

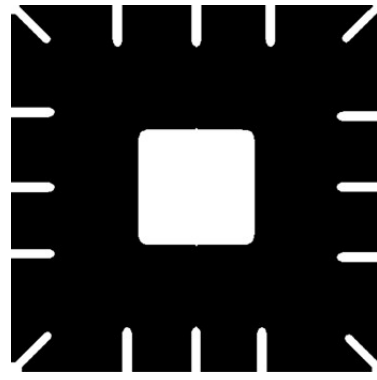
# Temel Morfolojiler



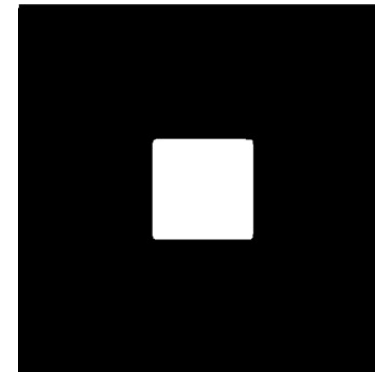
Orijinal görüntü



5 yarıçaplı  
diskle  
aşındırma



10 yarıçaplı  
diskle  
aşındırma



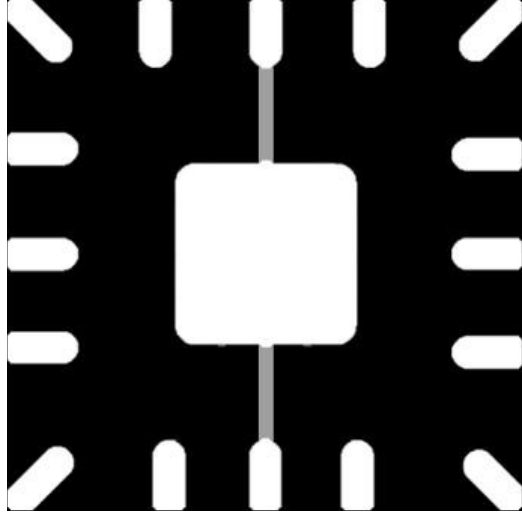
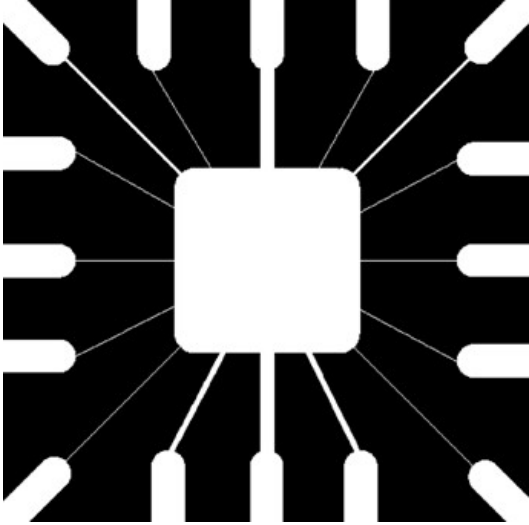
20 yarıçaplı  
diskle  
aşındırma

```
yapisal_eleman = strel('disk',20);  
imerode(A,yapisal_eleman);
```

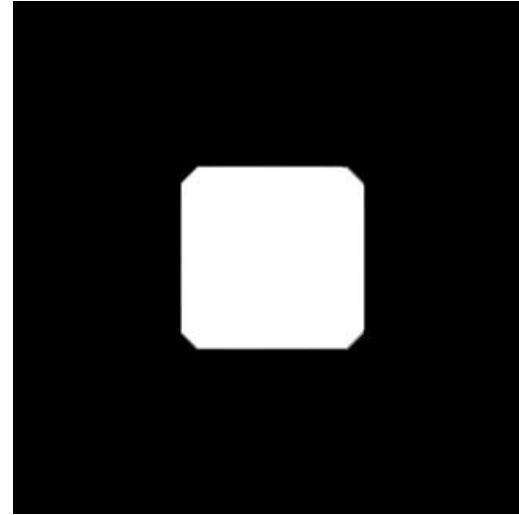
# Temel Morfolojiler

- **Closing (Kapama):** Önce dilation sonra erosion işlemi gerçekleştirilir.  
 $B \bullet S = (B \oplus S) \ominus S$ . Yakın noktaların birleştirilmesi, girinti ve çıkıntıların yok edilmesi gibi işlemlerde kullanılır.
- **Opening (Açma):** Closing in tam tersidir.  
 $B \circ S = (B \ominus S) \oplus S$   
Görüntü içindeki önemsiz küçük nesneler yok edilir.

# Opening



Yarıçap 5

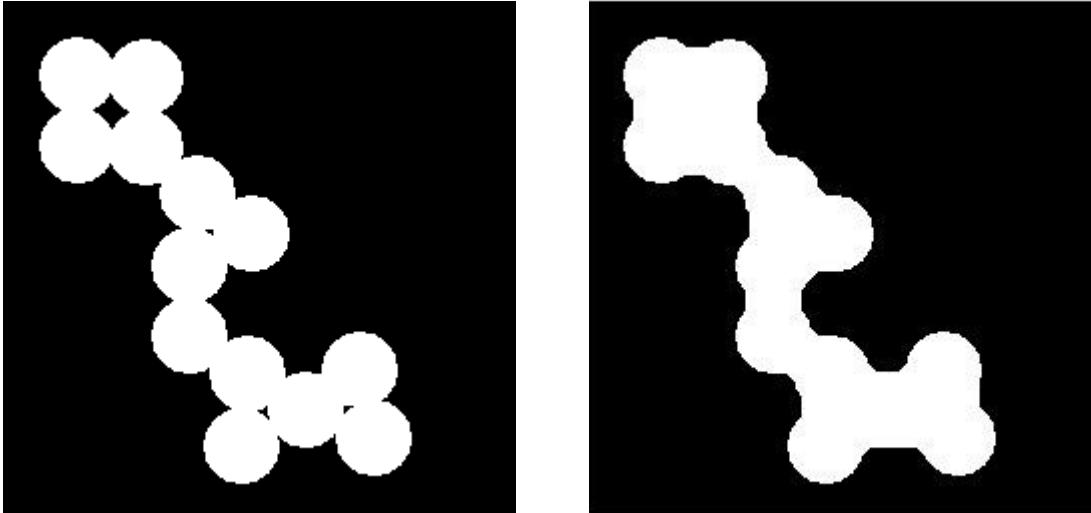


Yarıçap 20

Matlab komutu: **imopen(A,B)**

```
yapisa_eleman = strel('disk',5);  
imopen(A,yapisa_eleman);
```

# Closing



Matlab komutu: **imclose(A,B)**

```
yapisa_eleman = strel('disk',10);  
imclose(A,yapisa_eleman);
```



# Temel Morfolojiler

İkili  
Görüntü B

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

				1	1		
				1	1		
				1	1		

Erosion:  $B \ominus S$

	1	1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1				

Closing:  $B \bullet S = (B \oplus S) \ominus S$

			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	

Opening:  $B \circ S = (B \ominus S) \oplus S$

# Opening / Closing

Açma işlemi, bir görüntü içinde ön plandaki küçük nesneleri kaldırarak (genellikle koyu pikseller) arka plana yerleştirir.

Kapama işlemi ise, görüntü içinde ön plandaki küçük boşlukları kaldırarak arka plandaki küçük tepecikleri ön plana çıkarır.

# Opening / Closing

Disk şeklindeki bir yapısal elemanla bir görüntüye açma işlemi uygulandığında;

şekiller yumuşatılır, şekilleri birbirine bağlayan küçük parçalar silinir, küçük adacıklar/tepecikler silinir.

Disk şeklindeki bir yapısal elemanla bir görüntüye kapama işlemi uygulandığında,

şekiller yumuşatılır, dar kırılma noktaları ve uzun ince boşluklar kaynaştırılır, küçük boşluklar kaldırılır.

# Opening / Closing



Orijinal görüntü



Açma



Kapama



Kapama işlemi ve sonrasında  
açma işlemi

Bu örneklerde 20x20 kare yapısal eleman kullanılmıştır.

# Opening / Closing



Orijinal görüntü



Opening



Closing işlemi ve  
sonrasında  
opening işlemi

Burada 3x3 kare yapısal eleman kullanılmıştır.

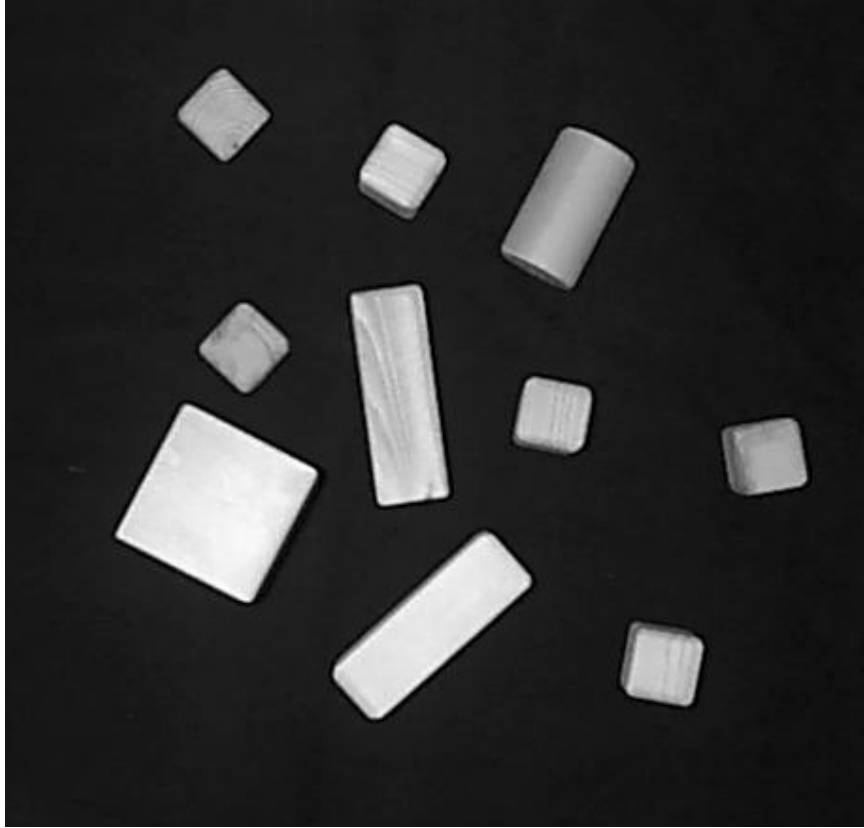
# Eşikleme (Thresholding)

- Arka plandan nesneleri ayırmak işlemidir (binarization).
- Pek çok nesne ya da gölge yüzeylerinden yansıttıkları ya da emdikleri ışık miktarıyla tanımlanabilirler.
- En basit segmentation işlemidir.
- İşlem sayısı az ve hızlı bir süreçtir.
- Özel donanımlarla gerçek zamanlı olarak gerçekleştirilebilir.

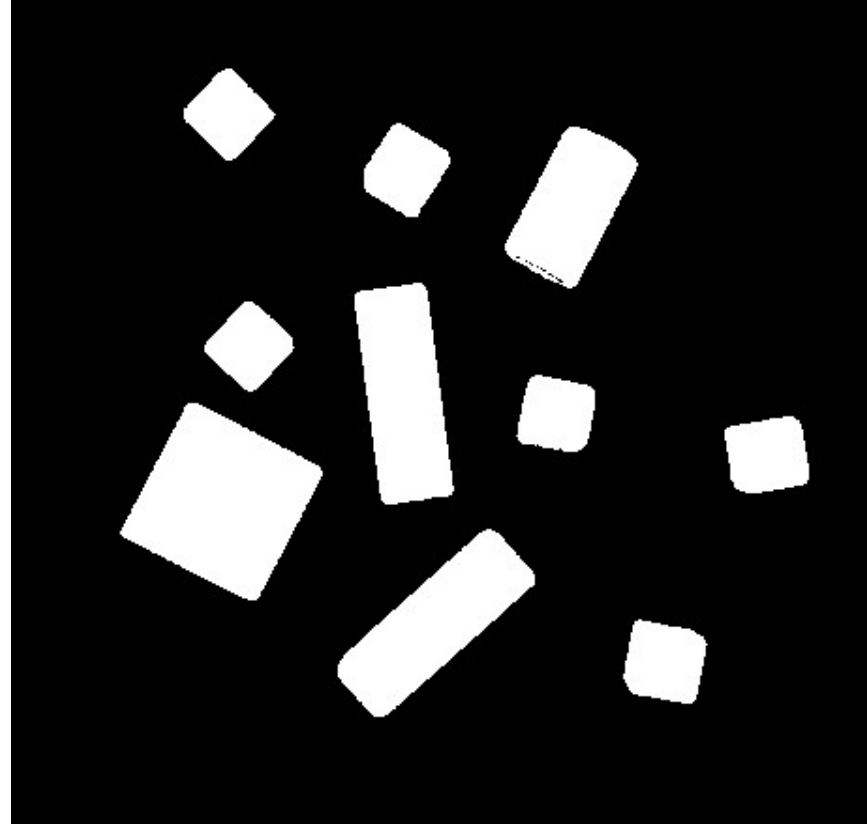
# Gri seviyeli Görüntülerde Eşikleme

- İkili görüntüler gri seviyeli görüntülerden eşikleme işlemi ile elde edilebilir.
- Eşikleme işlemi görüntünün bir kısım piksellerinin ön plan pikselleri olarak seçilip geri kalanlarının da arka plan pikselleri olarak kaldığı bir işlemdir.

# Gri seviyeli Görüntülerde Eşikleme



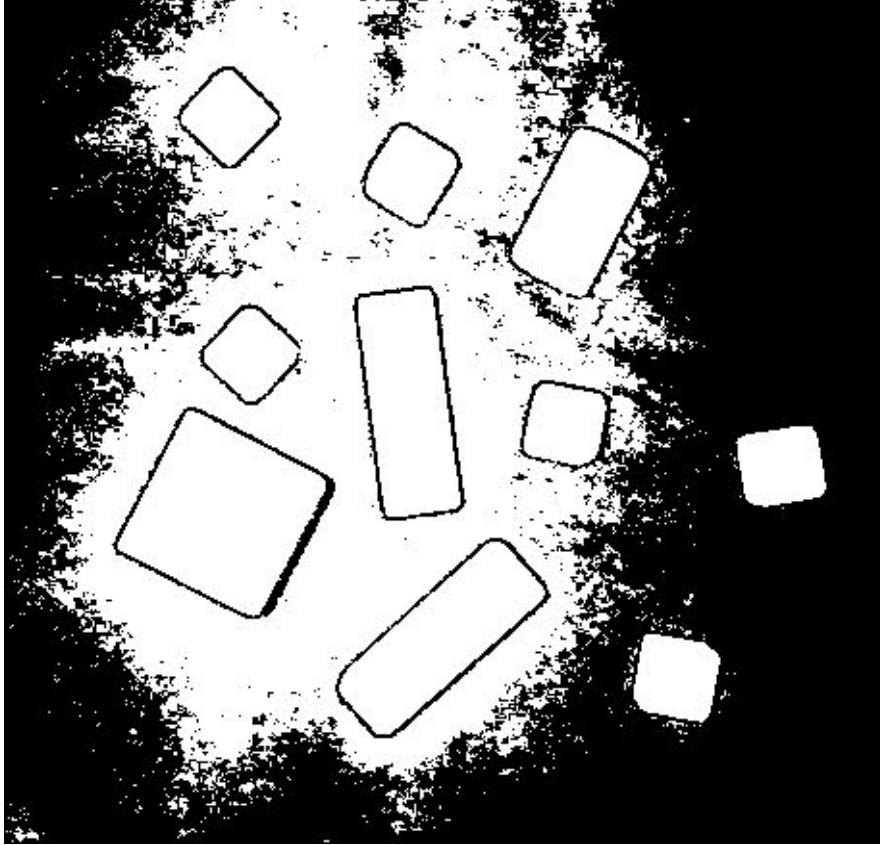
Original Görüntü



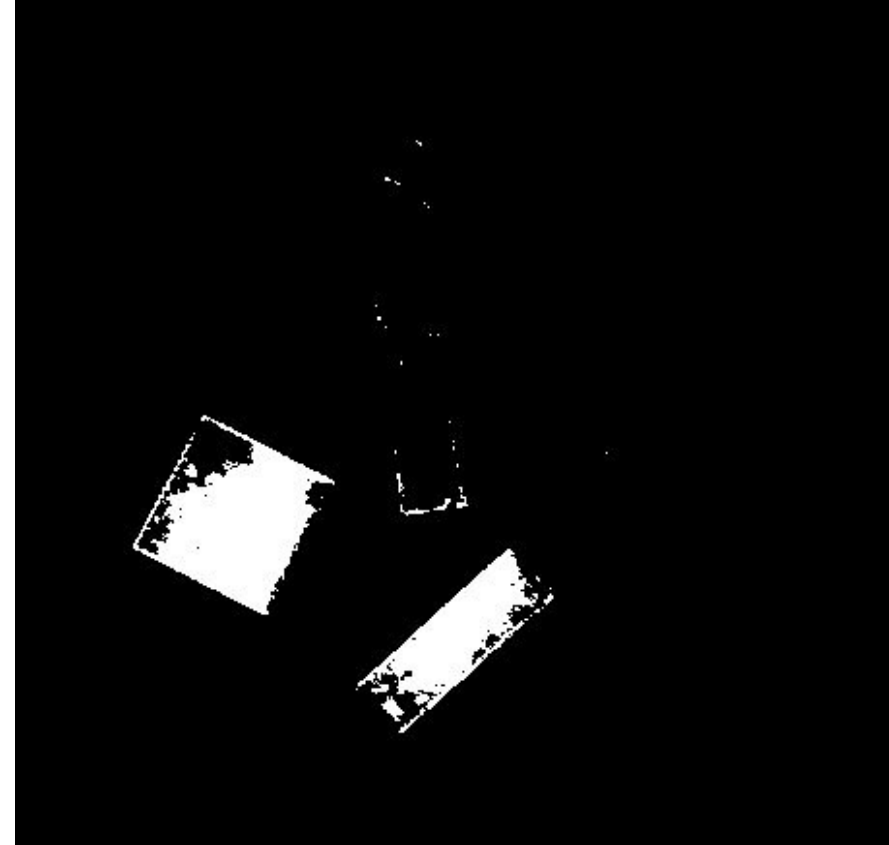
Sonuç



# Gri seviyeli Görüntülerde Eşikleme



Aşırı eşikleme



Az eşikleme

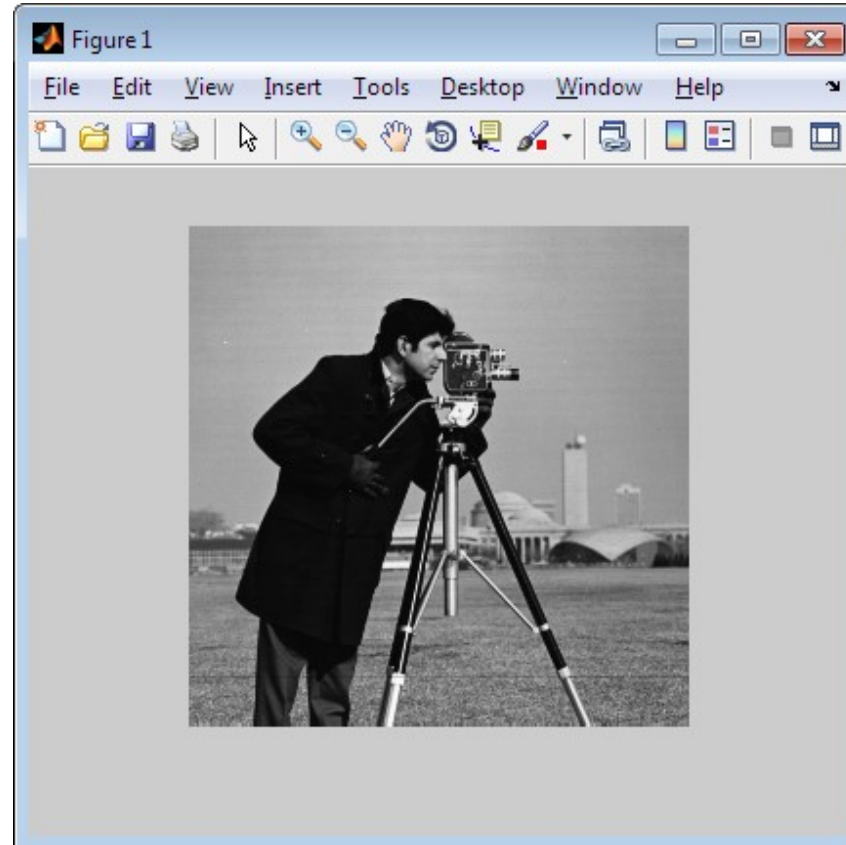
# Gri seviyeli Görüntülerde Eşikleme

- $f$  görüntüsünün tüm  $f(i,j)$  pikselleri taranır.
- Eğer  $f(i,j) \geq T$  ise bu piksel görüntü elemanı  $g(i,j)$  olarak adlandırılır, aksi halde arka plan pikselidir.
- Bu işlem için doğru eşik değerinin seçilmesi en kritik noktadır.
- Eşik değeri seçimi işleme göre gerçekleştirilebilir ya da bazı yöntemlerin sonucuna göre belirlenebilir.

$$\begin{aligned} g(i, j) &= 1 && \text{for } f(i, j) \geq T \\ &= 0 && \text{for } f(i, j) < T \end{aligned} \quad (5.2)$$

# Gri seviyeli Görüntülerde Eşikleme

```
imge = imread('cameraman.tif');  
imshow(imge);
```



# im2bw ile Eşikleme

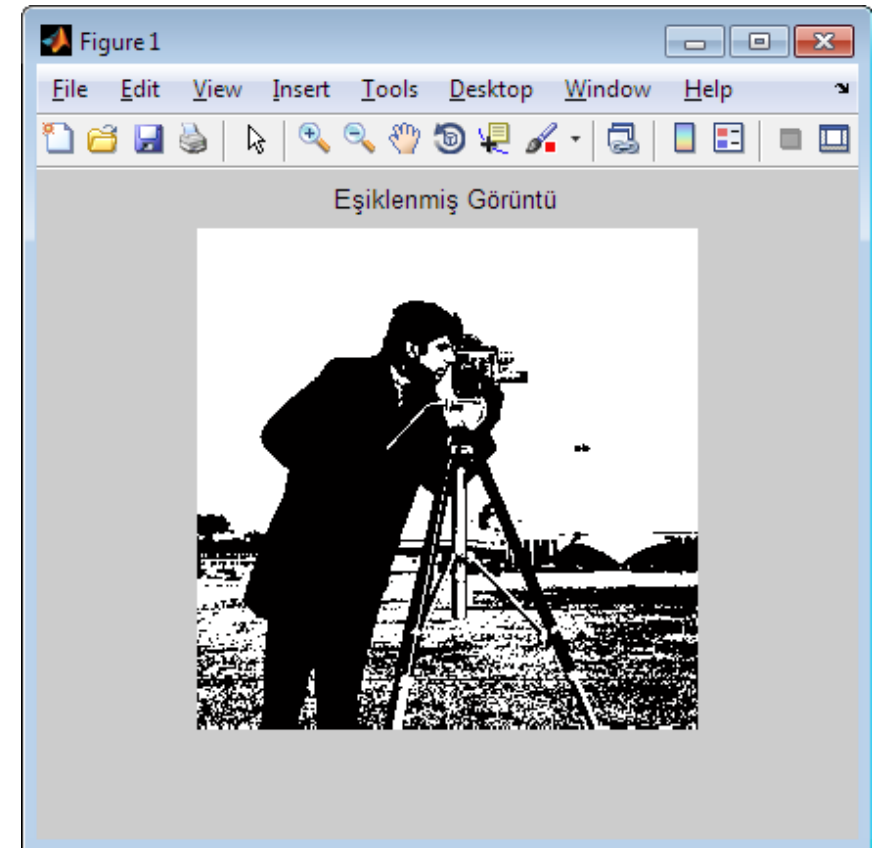
- im2bw fonksiyonu parametre olarak bir eşikleme sınır değeri alır.
- Sınır değerini ister deneme yanılma ile, istersek histogram inceleme ile bulunabilir.
- Optimum bir eşik değeri için görüntünün histogramını incelemek daha faydalı olacaktır.

# im2bw ile Eşikleme

- Önce elimizdeki gri görüntüye 0.5 sınır değeri ile eşikleme yapalım.

```
imge2 = im2bw(imge,0.5);
```

```
imshow(imge2),title('Eşiklenmiş Görüntü')
```



# Graythresh Fonksiyonu ile Eşik Değerinin Alınması

- Matlab da tanımlı graythresh fonksiyonunu kullanarak elimizdeki görüntü için otomatik olarak ideal bir eşik sınırı elde etmeye çalışalım.
- Graytrash histogram bazlı bir yöntemdir. Bu yöntemde algoritma önce görüntünün histogramını hesaplar, ardından otomatik bir threshold değeri döner.

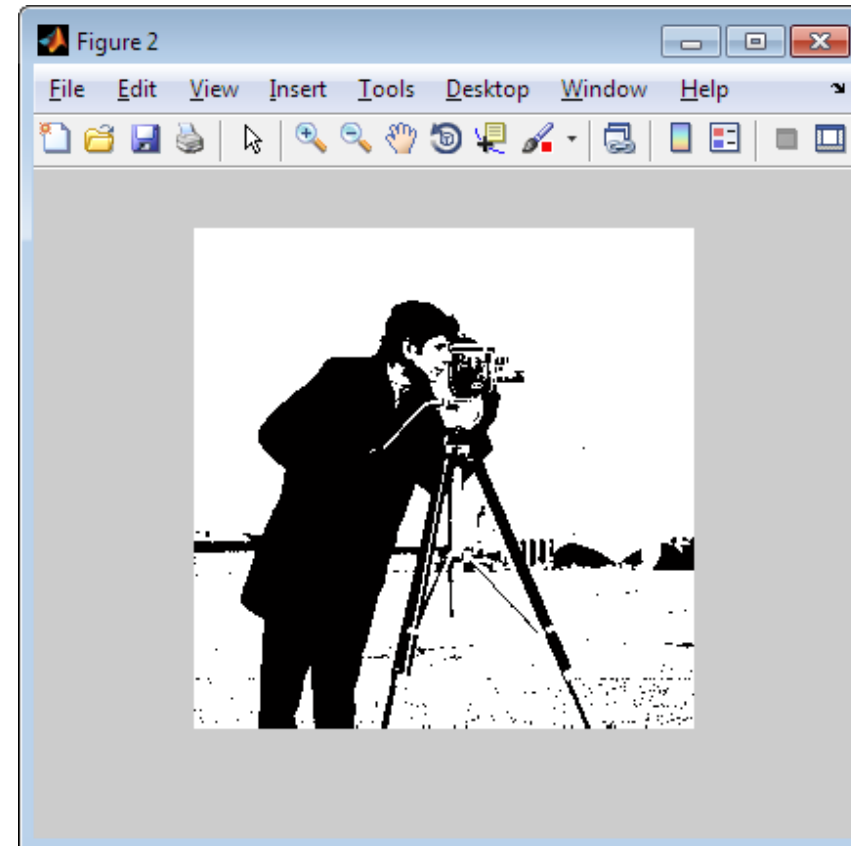
# Graythresh Fonksiyonu ile Eşik Değerinin Alınması

Esikdegeri=graythresh(imge)

- Esikdegeri=0.3451 belirlenir. (Daha önce alınan 'cameraman.tif' görüntüsü için)
- Buradan çıkar değer bizim belirlediğimiz yaklaşık 0.29 değer uzaklığında.
- Otomatik bulunan bu değerle eşikleme yapılırsa.

# Graythresh Fonksiyonu ile Eşik Değerinin Alınması

```
imge2 = im2bw(imge,esikdegeri);  
Figure, imshow(imge2)
```





```
I=imread('cameraman.tif');  
for i=1:size(I,1)  
    for j=1:size(I,2)  
        if I(i,j)>120  
            I2(i,j)=255;  
        else I2(i,j)=0;  
        end  
    end  
end  
imshow(I), figure(), imshow(I2)
```

# Histogram

- Eşik değeri seçimi kullanıcı tarafından belirlenebilir ancak görüntü analiz işlemlerinin otomatik olarak gerçekleşmesi gerekir.
- Bir eşik seçiminin temel ilkesi gri seviyeli görüntünün histogramının kullanılmasıdır.

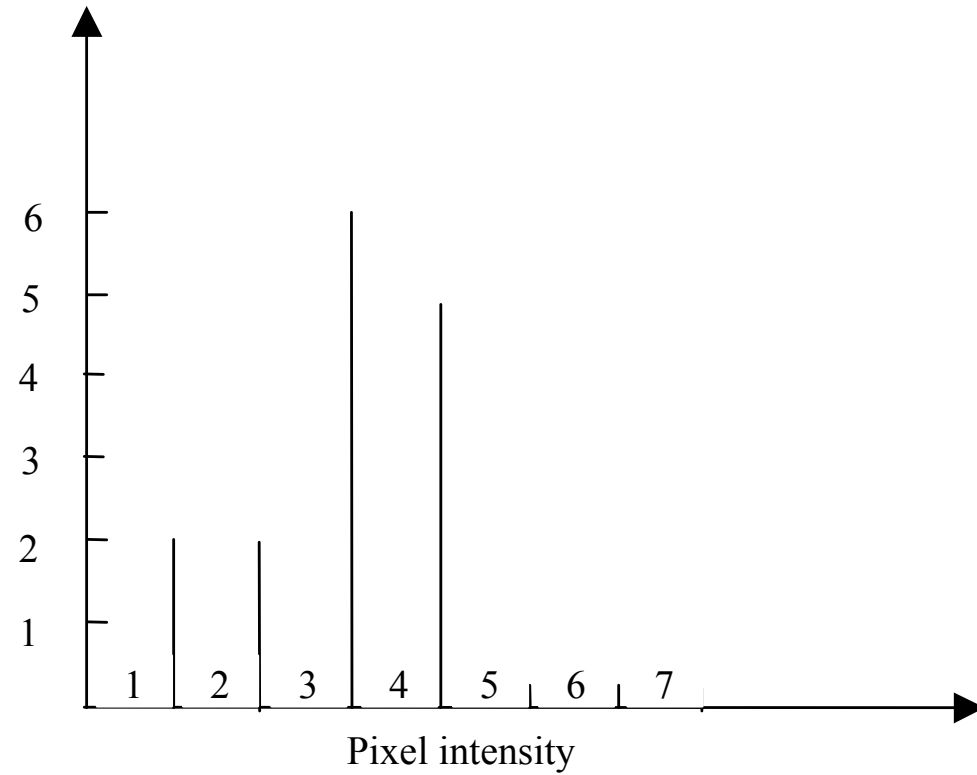
# Histogram

- Görüntü Histogramı bir görüntünün yoğunluk profiline bakmak için kullanılan değerli bir araçtır. Histogram, zıtlık ve görüntünün yoğunluk dağılımı ile ilgili bilgi sağlar. Görüntü histogramı piksel yoğunluğunun basit bir çubuk grafiğidir. Piksel yoğunluğu x eksenini boyunca ve her bir yoğunluktan kaç tane olduğu da y eksenini boyunca gösterilir.
- Koyu görüntüler sol tarafa doğru yığılmış bir histograma sahiptir. Parlak görüntüler ise sağ tarafa doğru yığılmış bir histograma sahiptir. İdeal bir görüntüde histogram boyunca düzenli bir dağılım vardır.

# Histogram

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

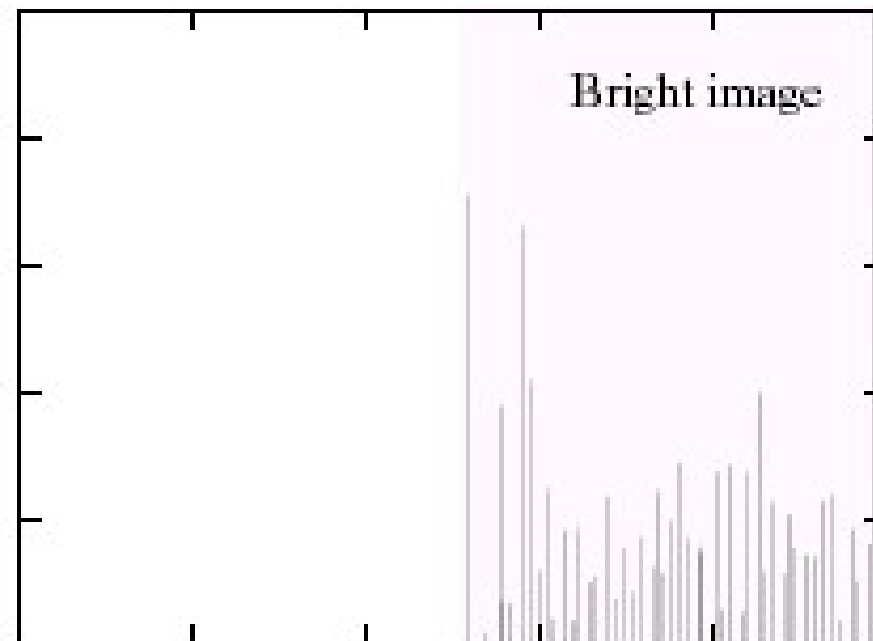
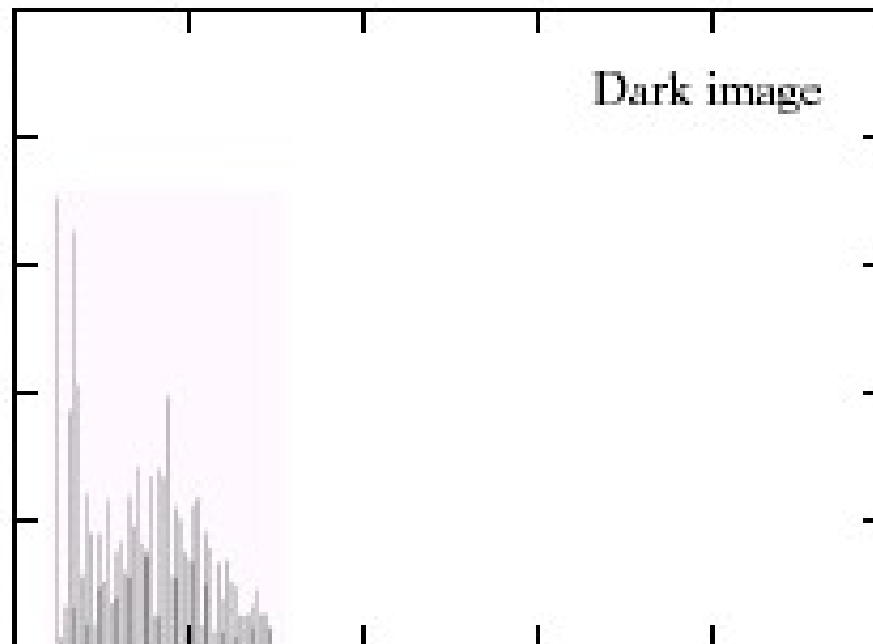
Image



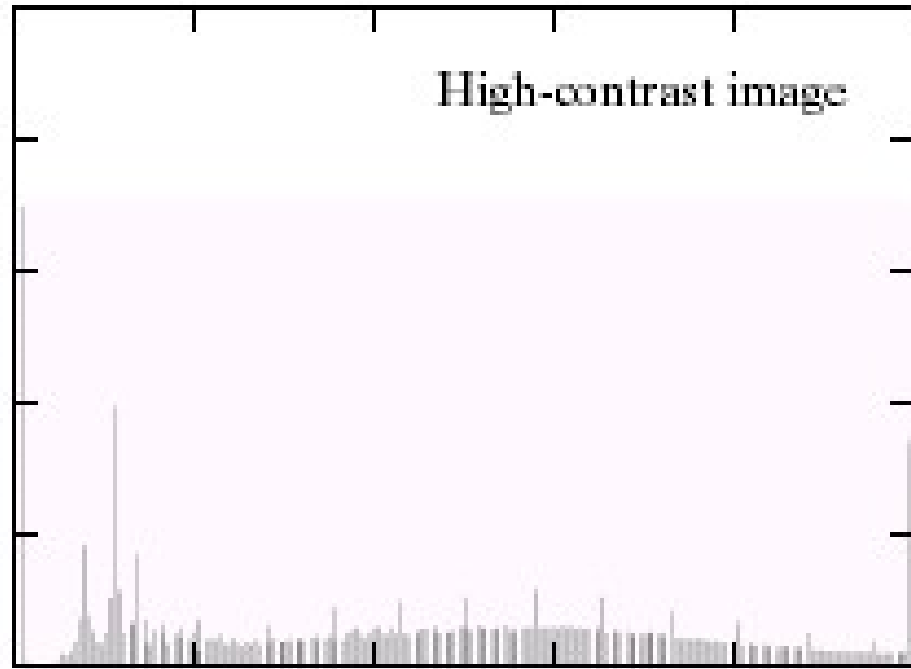
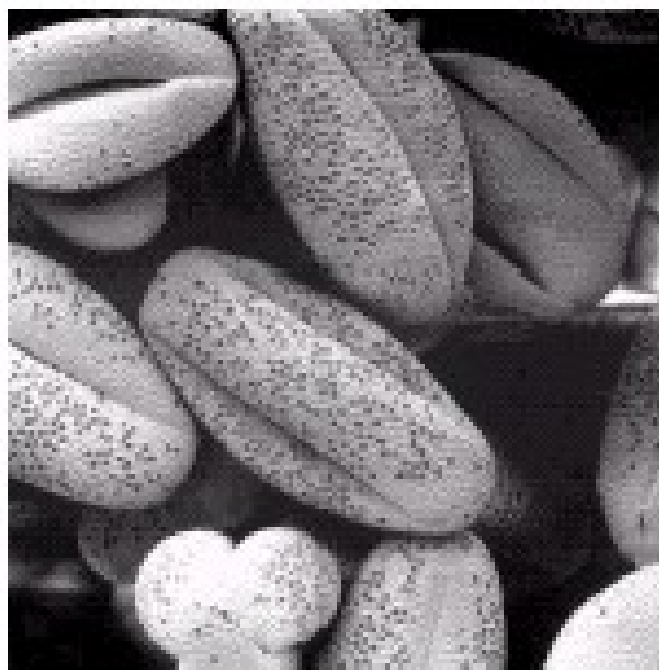
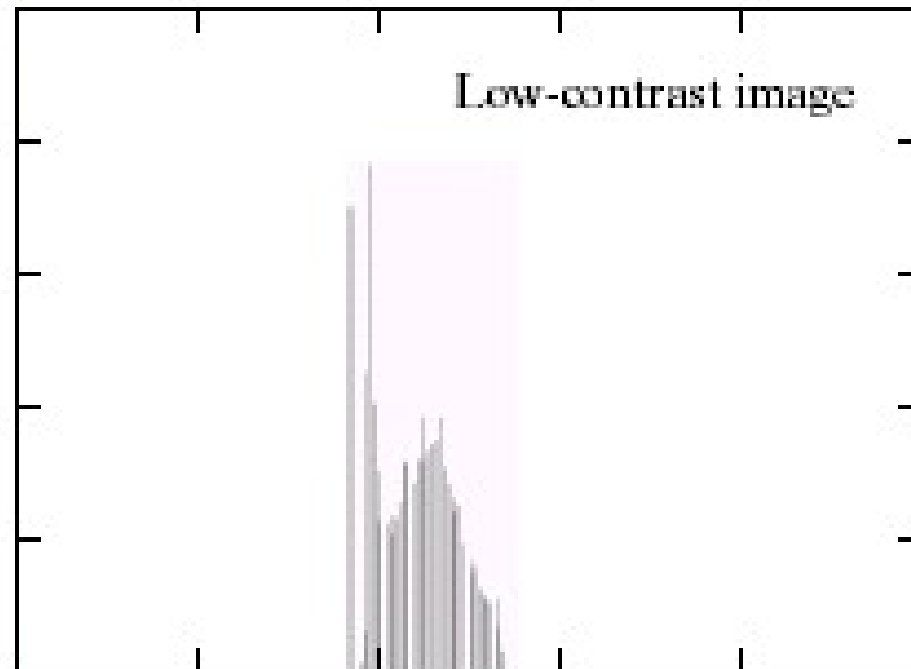
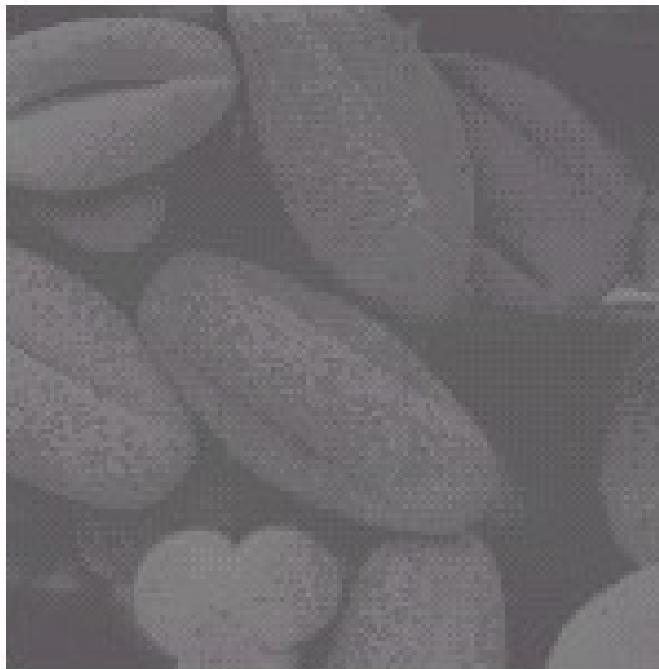
# Histogram

- Bir görüntünün kontrastı parlak ve koyu piksellerin dağılımıdır. Düşük zıtlıklı gri seviyeli görüntüler daha koyu, daha parlak veya daha gridir. Düşük zıtlıklı bir görüntünün histogramında pikseller sağa, sola veya ortaya dağılmıştır. Ayrıca histogram çubukları birlikte sıkıca kümelenmiştir ve bütün olası piksel değerlerinin küçük bir kısmını kullanır.
- Yüksek zıtlıklı görüntüler hem koyu, hem de açık bölgelere sahiptir. Yüksek zıtlıklı görüntü problemi görüntünün geniş koyu bölgeye ve geniş açık bölgelere sahip olmasıdır. Güneşli bir günde bir pencerenin önünde duran birinin resmi yüksek zıtlıklıdır. Yüksek zıtlıklı görüntülerin histogramları iki büyük tepeye sahiptir. Bir tepe düşük bölgenin ortasında, diğeri yüksek bölgenin ortasında yer alır.

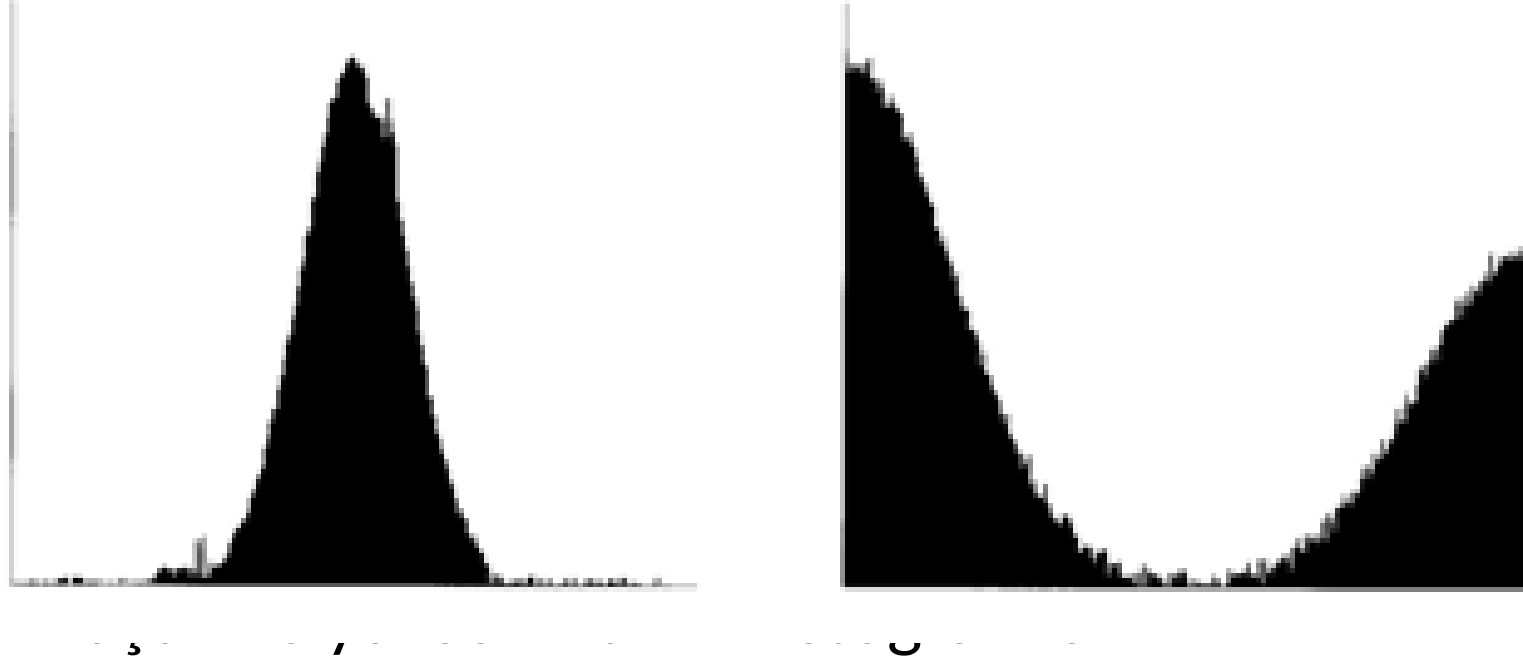
# Histogram



# Histogram



# Histogram

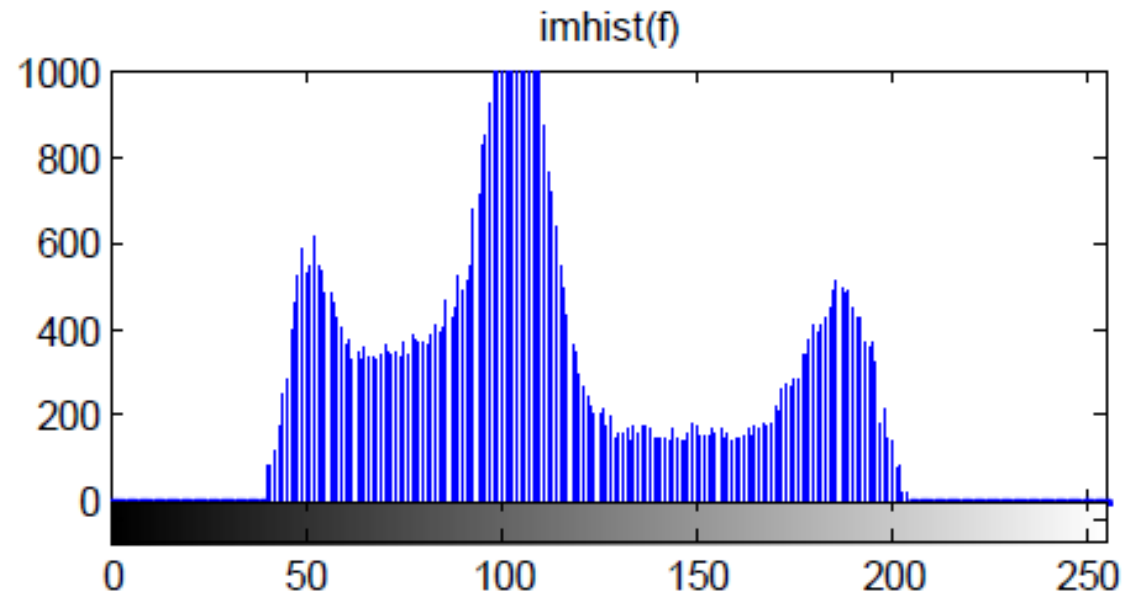


Düşük ve yüksek zıtlıklı histogramlar



# Histogram

```
imge = imread('cameraman.tif');  
figure,imhist(imge)
```



# Temel Ton Dönüşümleri

- Görüntünün Negatifi
- Logaritmik Dönüşüm

# Görüntünün Negatifi

- Yoğunluk seviyeleri tersine çevrilir.
- Beyaz ya da siyah ayrıntıların iyileştirilmesi gerektiğinde uygundur.
- Tüm piksel değerleri maksimum ton değerinden çıkarılarak mutlak değeri alınır.
- $[0, L-1]$  aralığındaki bir imgenin ( $r$ : orijinal değer) negatifi için şekilde gösterilen fonksiyon kullanılır:  **$s = L - 1 - r$**

# Görüntünün Negatifi

Orjinal Görüntü



Negatif Görüntü



# Görüntünün Negatifi

```
r2 = imcomplement(r1);
```

```
figure,imshow(r2);
```

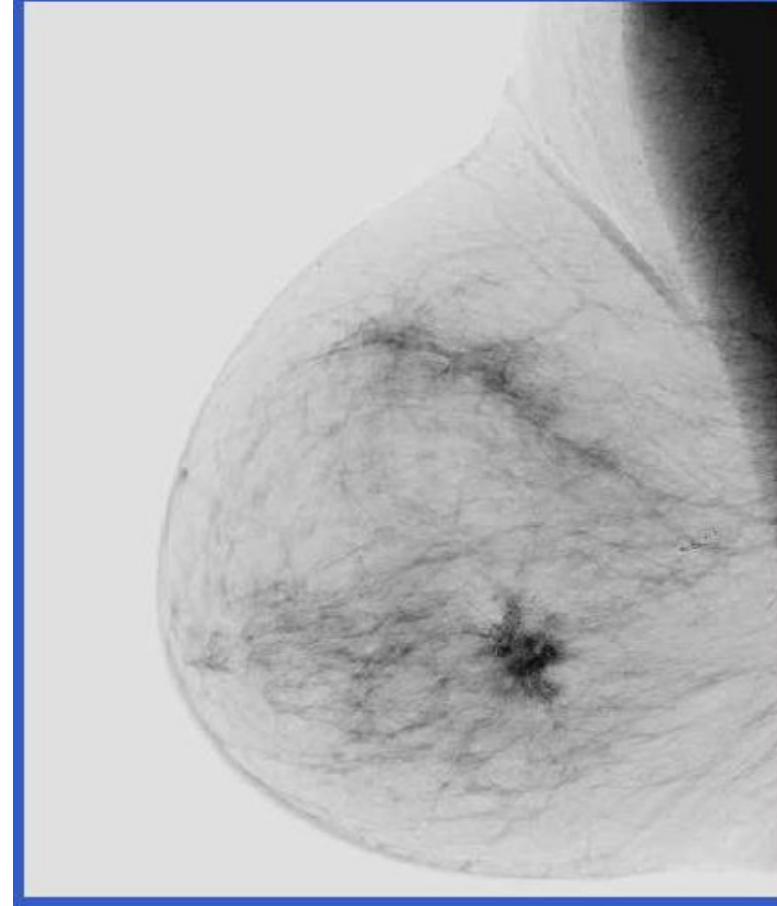
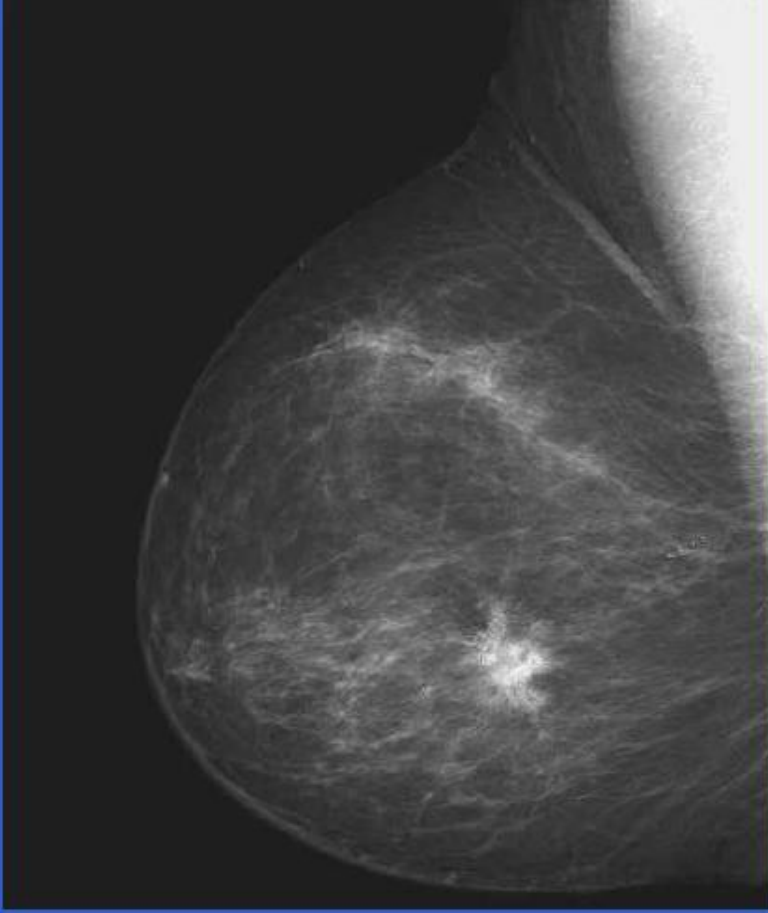
**veya**

```
r2 = imadjust(r1,[0 1], [1 0]);
```

```
figure,imshow(r2);
```



# Görüntünün Negatifi



# Logaritmik Dönüşüm

- Genelde büyük değerleri görüntülemek için kullanılır. Örneğin, Fourier dönüşümü sonrasında elde edilen değerler 0 ile  $10^6$  aralığında çıkar. Bu değerleri gereken aralık içine çekmek için kullanılabilir.

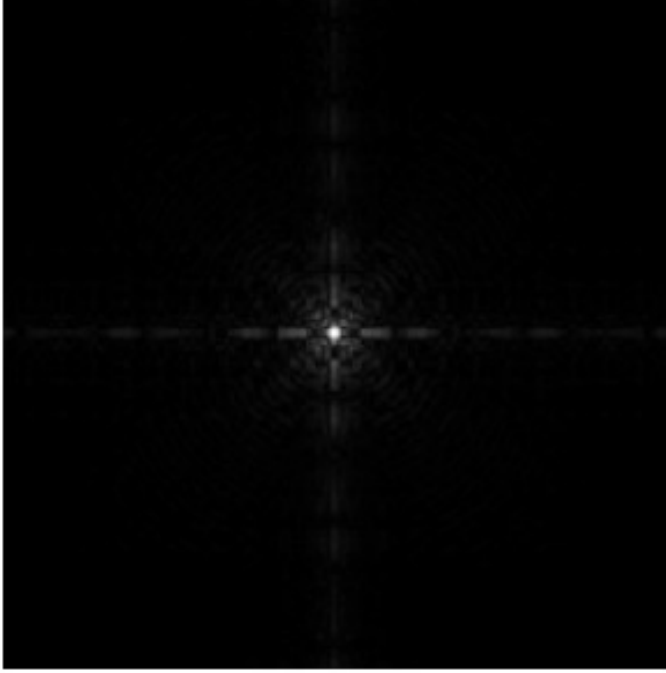
$$s = c \log (1 + r)$$

- c: sabit,  $r \geq 0$

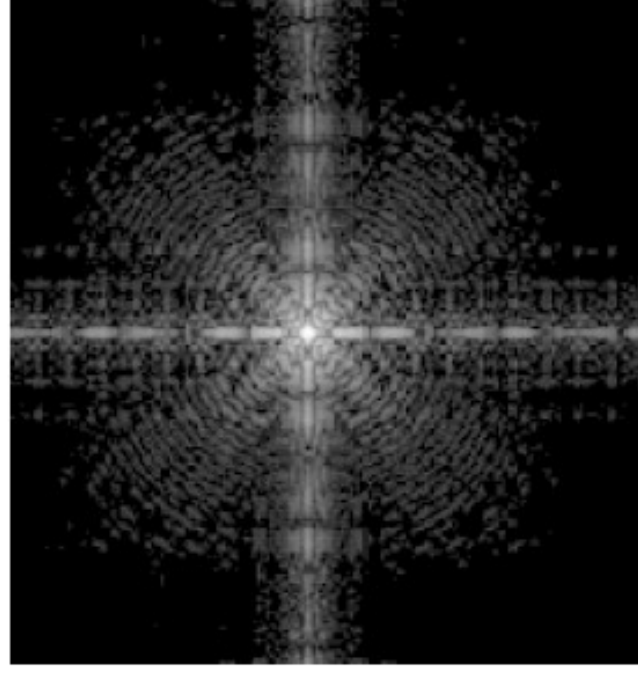
- r: orjinal değer, c genelde 1 alınır.

# Logaritmik Dönüşüm

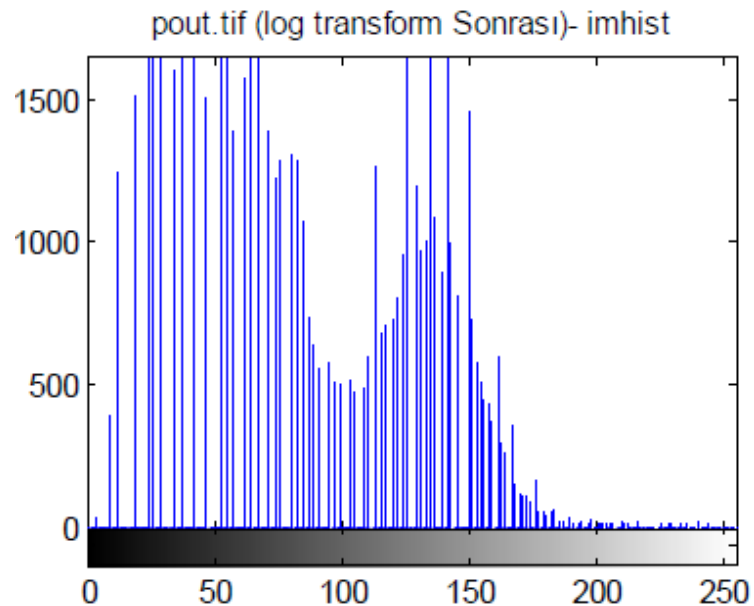
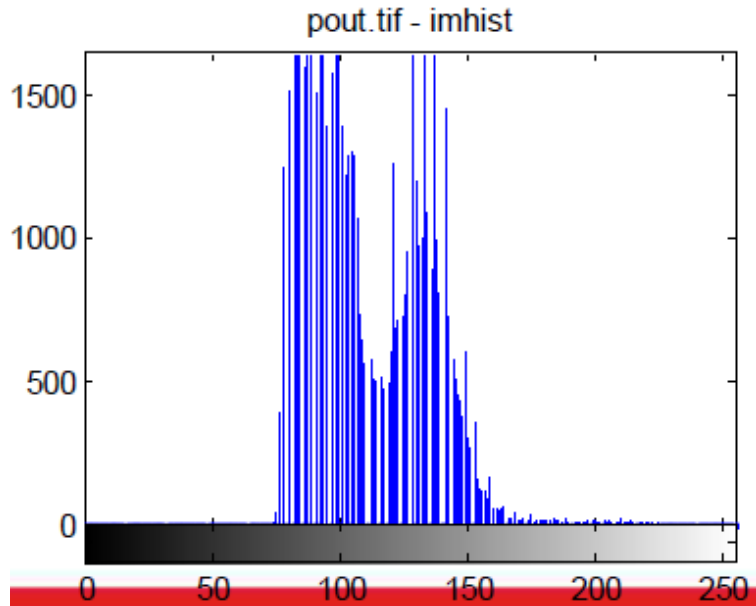
Orjinal Görüntü



Logaritmik dönüşüm Görüntüsü







pout.tif - Orjinal imge



pout.tif (log transform Sonrası)



```
I = imread('pout.tif');  
figure,imshow(I); title('pout.tif - Orjinal imge')  
figure,imhist(I), title('pout.tif - imhist')  
g=im2uint8(mat2gray(log(1+double(I))));  
figure,imshow(g); title('pout.tif (log transform Sonrası)')  
figure,imhist(g),title('pout.tif (log transform Sonrası)- imhist')
```

# Görüntü İyileştirme

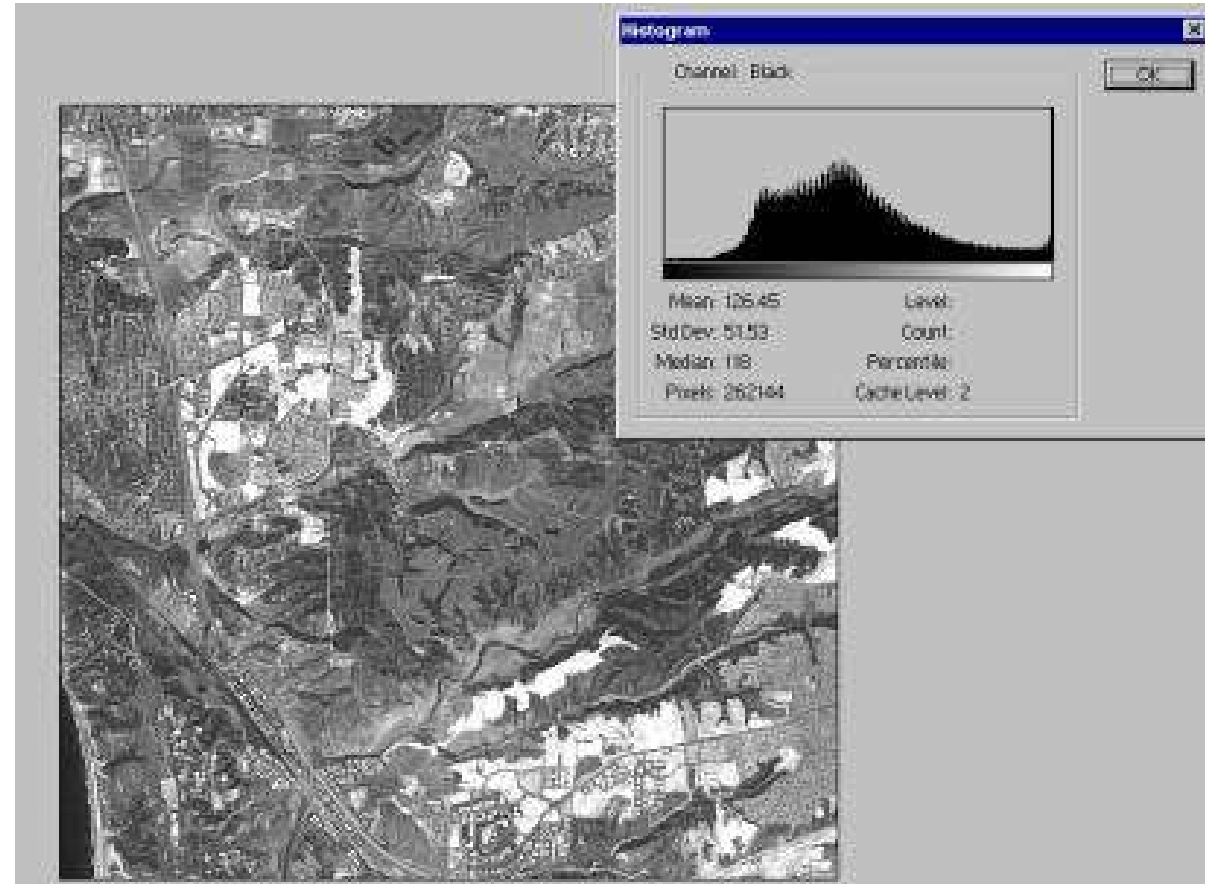
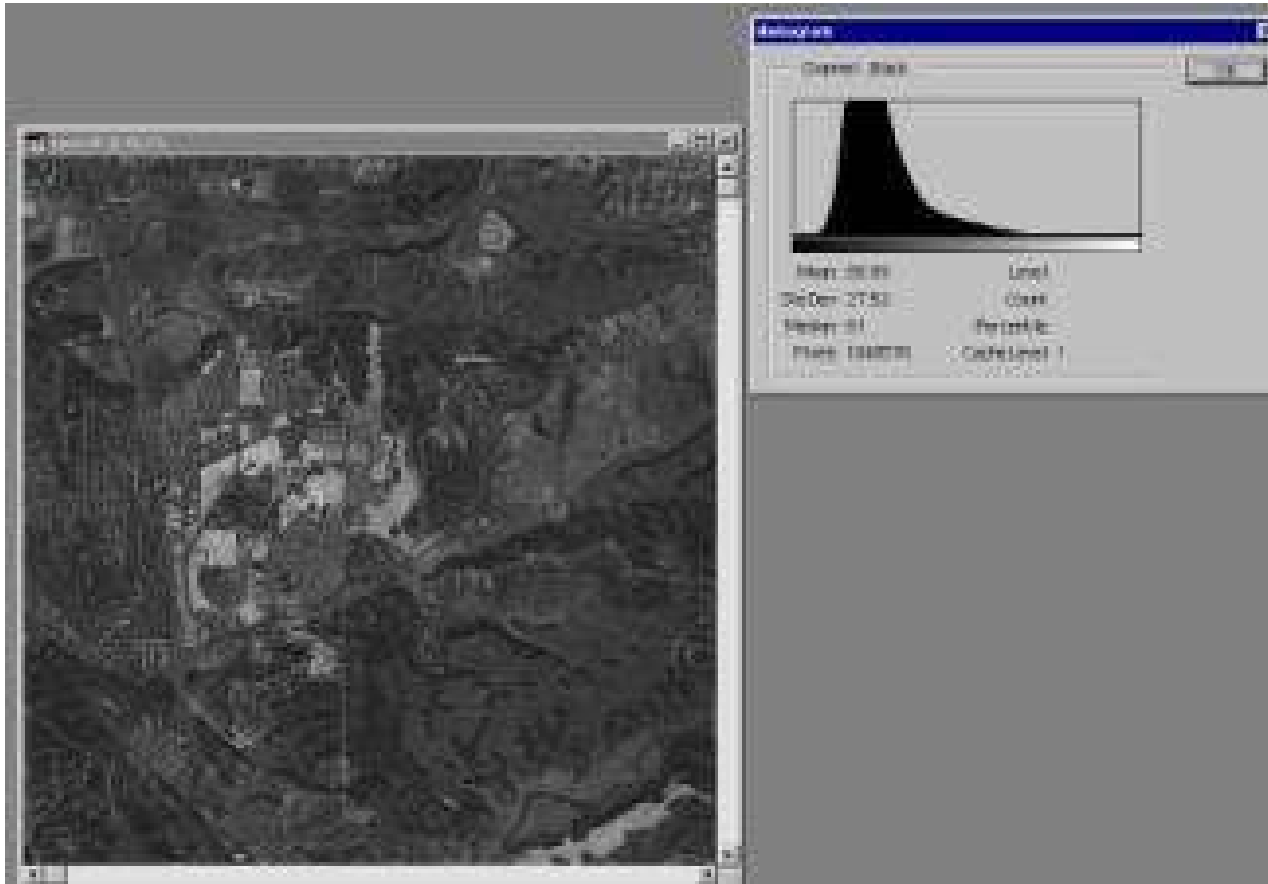
- Çok koyu ya da çok açık görüntüler üzerinde uygulanır. Sıkışık haldeki histogram açılır.
- Histogram Germe
- Histogram Eşitleme

# Histogram Germe

- En basit kontrast zenginleştirme yöntemidir.
- Histogramdan orijinal görüntünün minimum ve maksimum değerleri belirlenir ve bu aralık dinamik aralığın tamamına yayılacak şekilde aşağıdaki eşitlik kullanılarak dönüştürülür.

$$j = \frac{YENİ_{mak} - YENİ_{min}}{ESKİ_{mak} - ESKİ_{min}} (i - ESKİ_{min}) + YENİ_{min}$$

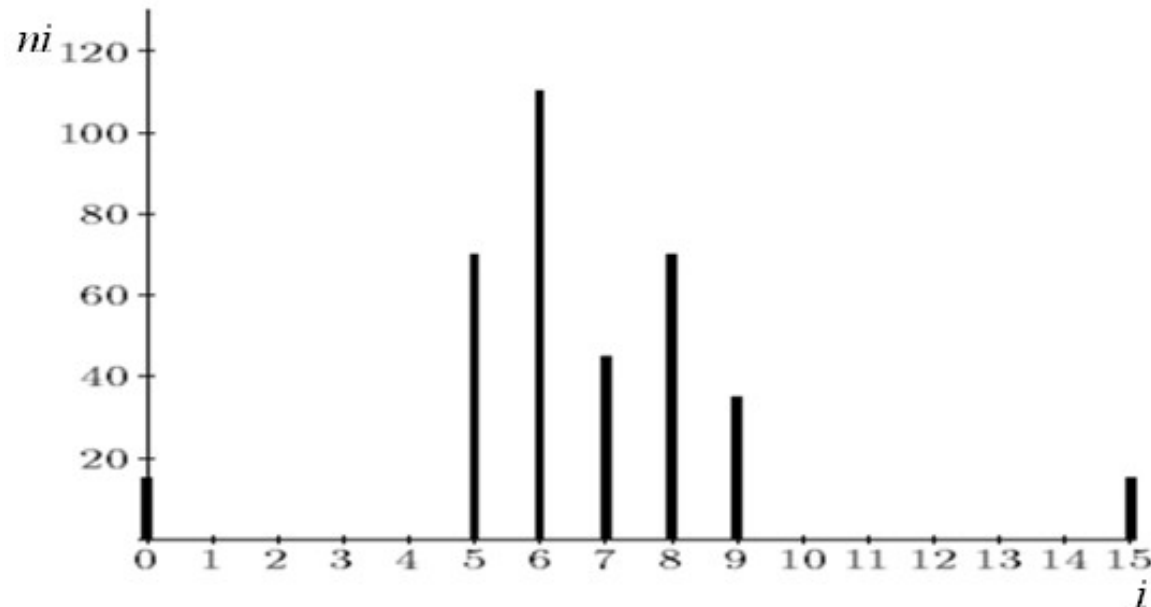
# Histogram Germe



# Histogram Germe

- Histogramı ve tablosu aşağıda verilen bir görüntüye bakalım. Burada;  $n_i$ ,  $i$ . gri seviyedeki piksel sayısı değeridir. Biz bu gri seviye değerlerini, lineer fonksiyonu uygulayarak, orijinal aralığın dışına yayabiliriz.

Grey level $i$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$n_i$	15	0	0	0	0	70	110	45	70	35	0	0	0	0	0	15

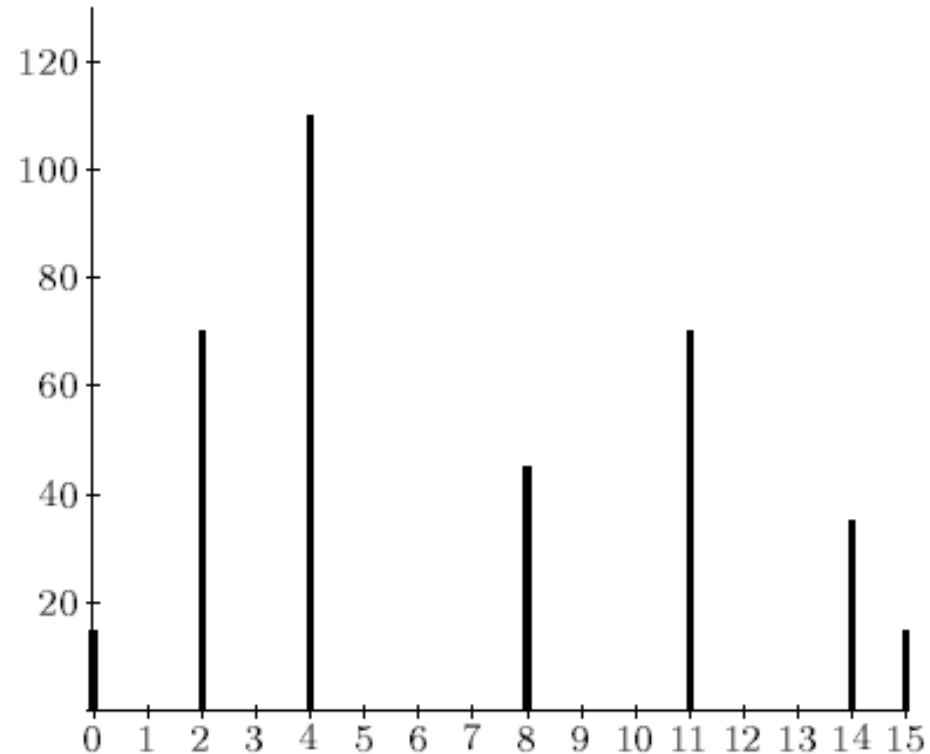


# Histogram Germe

- 5-9 aralığındaki gri değerleri 2-14 aralığına çekersek,

$$j = \frac{14 - 2}{9 - 5}(i - 5) + 2$$

$i$	5	6	7	8	9
$j$	2	5	8	11	14



# Histogram Germe

```
I = imread('pout.tif');  
imshow(I);
```



```
J = imadjust(I);  
figure  
imshow(J)
```

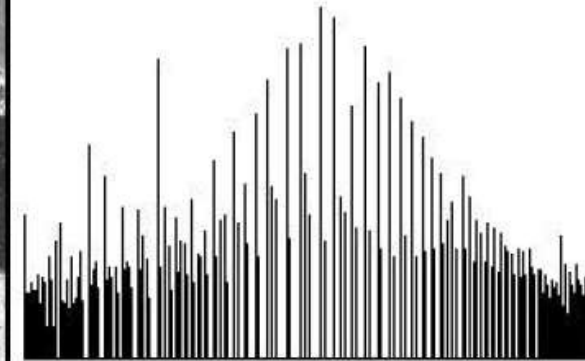
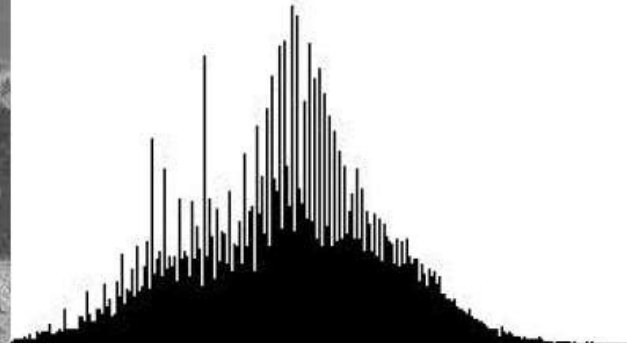


# Histogram Eşitleme

- Amaç, çıktı görüntü histogramının **uniform** bir dağılımda olmasını yani her bir parlaklık seviyesi için yaklaşık aynı sayıda piksel bulunmasını amaçlar.
- Histogram eşitleme renk değerleri düzgün dağılımlı olmayan resimler için uygun bir görüntü iyileştirme metodudur.



# Histogram Eşitleme



# Histogram Eşitleme

- $p(r_k)$ ,  $k$ . ton değerinin görüntü içinde hangi oranda temsil edildiğini gösterir:

$$p_r(r_k) = \frac{n_k}{n} \quad k = 0, 1, 2, \dots, L - 1$$

- $n_k$ ,  $k$ . tondan görüntü içinde kaç adet olduğu ve
- $n$ , toplam piksel sayısıdır.

# Histogram Eşitleme

- Önce kümülatif olasılık fonksiyonu  $s_k$  hesaplanır.

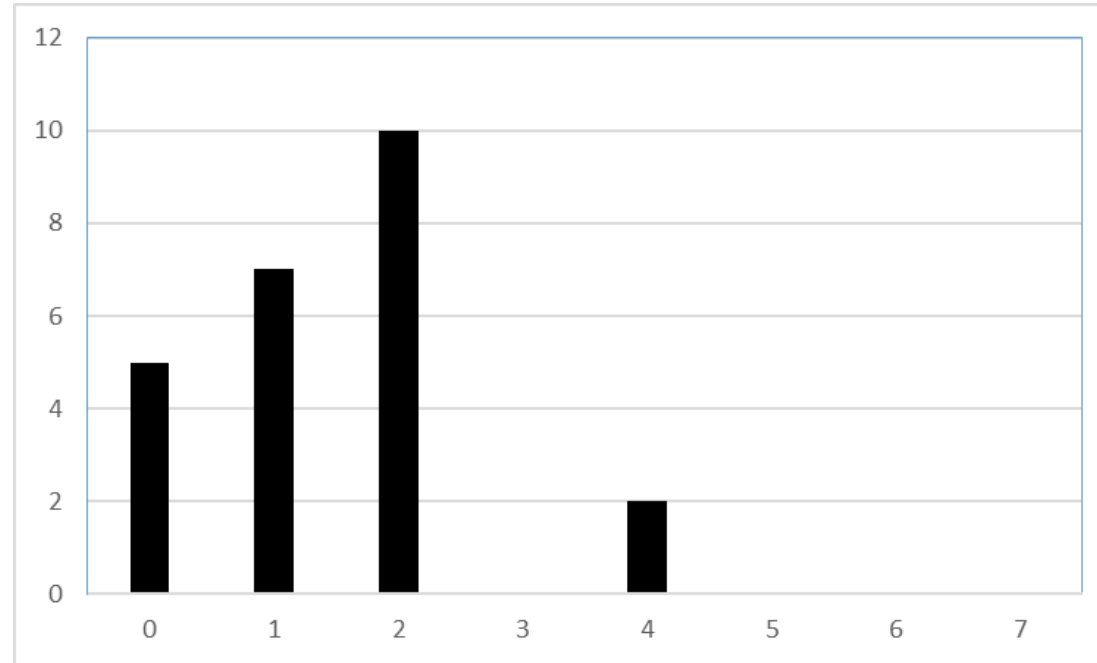
$$\begin{aligned} s_k &= T(r_k) = \sum_{j=0}^k p_r(r_j) \\ &= \sum_{j=0}^k \frac{n_j}{n} \quad k = 0, 1, 2, \dots, L - 1. \end{aligned}$$

- Daha sonra da ters dönüşüm yapılarak hangi renk tonu yerine ne geleceği hesaplanır.
- Daha sonra da ters dönüşüm yapılarak hangi renk tonu yerine ne geleceği hesaplanır.

$$r_k = T^{-1}(s_k) \quad \text{veya} \quad r_k = T^{-1}(s_k) = (L - 1) * T(r_k), \quad k = 0, 1, 2, \dots, L - 1$$

# Histogram Eşitleme

- Toplam 24 piksel ve 8 gri tona sahip bir görüntü üzerinde uygularsak,



# Histogram Eşitleme

$$r = 0, \quad P[0] = \frac{5}{24} = 0,208, \quad T[0] = 0,208, \quad T^{-1}[0] = 0,208 * 7 \cong 2, \quad 0 \rightarrow 2$$

$$r = 1, \quad P[1] = \frac{7}{24} = 0,291, \quad T[1] = 0,5, \quad T^{-1}[1] = 0,5 * 7 \cong 4, \quad 1 \rightarrow 4$$

$$r = 2, \quad P[2] = \frac{10}{24} = 0,416, \quad T[2] = 0,916, \quad T^{-1}[2] = 0,916 * 7 \cong 6, \quad 2 \rightarrow 6$$

$$r = 3, \quad P[3] = 0, \quad T[3] = 0,916, \quad T^{-1}[3] = 0,916 * 7 \cong 6, \quad 3 \rightarrow 6$$

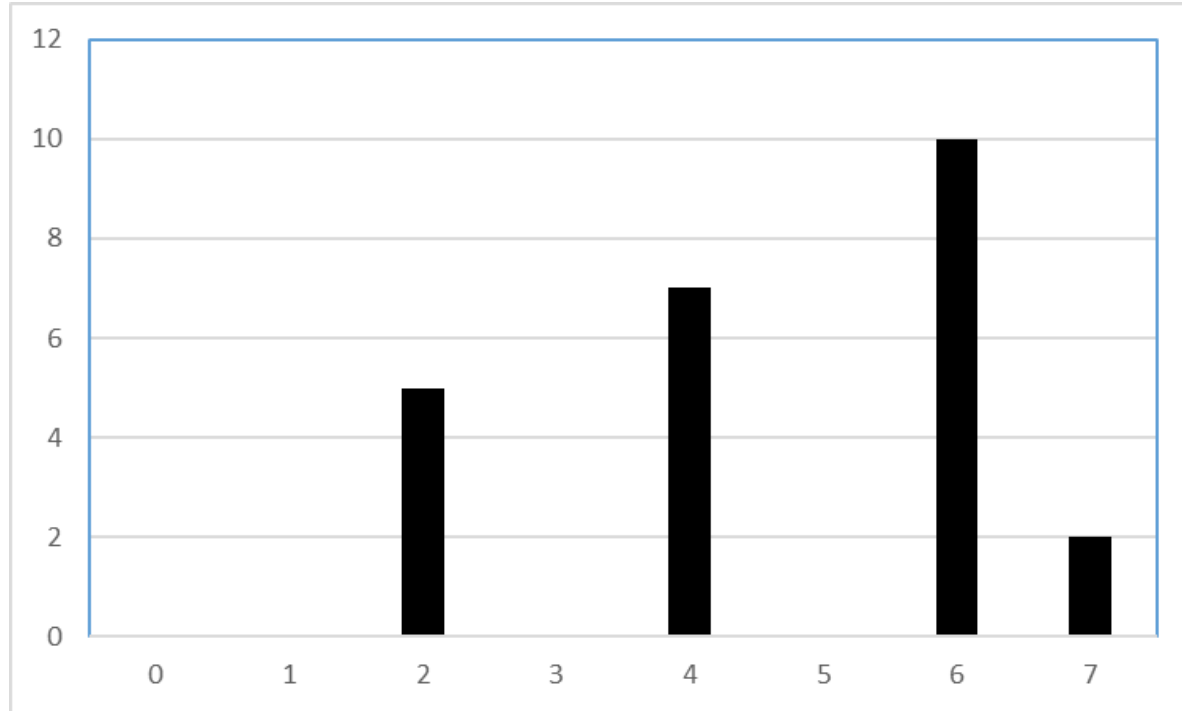
$$r = 4, \quad P[4] = \frac{2}{24} = 0,083, \quad T[4] = 1, \quad T^{-1}[4] = 1 * 7 = 7, \quad 4 \rightarrow 7$$

$$r = 5, \quad P[5] = 0, \quad T[5] = 1, \quad T^{-1}[5] = 1 * 7 = 7, \quad 5 \rightarrow 7$$

$$r = 6, \quad P[6] = 0, \quad T[6] = 1, \quad T^{-1}[6] = 1 * 7 = 7, \quad 6 \rightarrow 7$$

$$r = 7, \quad P[7] = 0, \quad T[7] = 1, \quad T^{-1}[7] = 1 * 7 = 7, \quad 7 \rightarrow 7$$

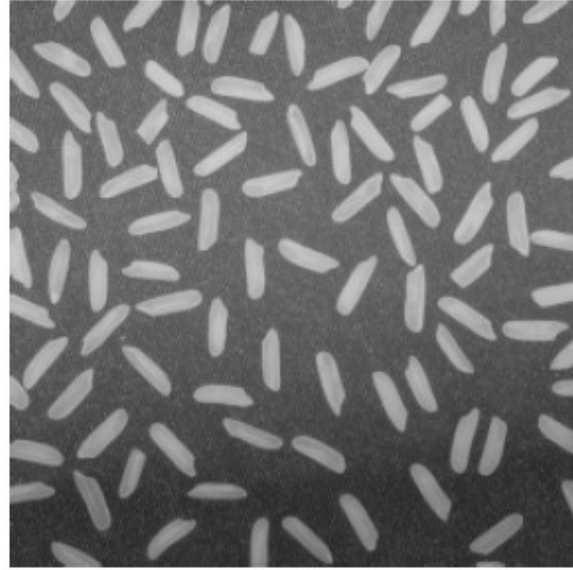
# Histogram Eşitleme



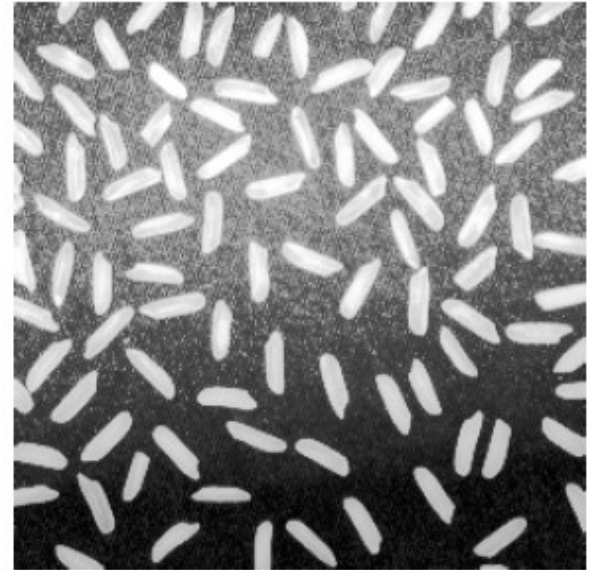
# Histogram Eşitleme

```
f=imread('rice.png');  
figure,imshow(f);title('Orjinal Resim')  
figure;  
imhist(f);title('Orjinal Histogram')  
ylim('auto');  
g=histeq(f,256);  
figure,imshow(g);  
title('Eşitleme Sonrası Resim')  
figure;imhist(g);  
title('Eşitleme Sonrası Histogram ')  
ylim('auto');
```

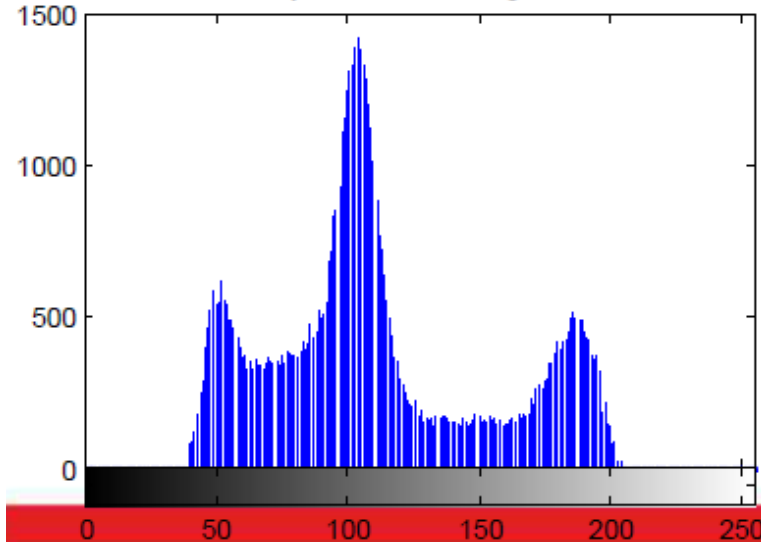
Orjinal Resim



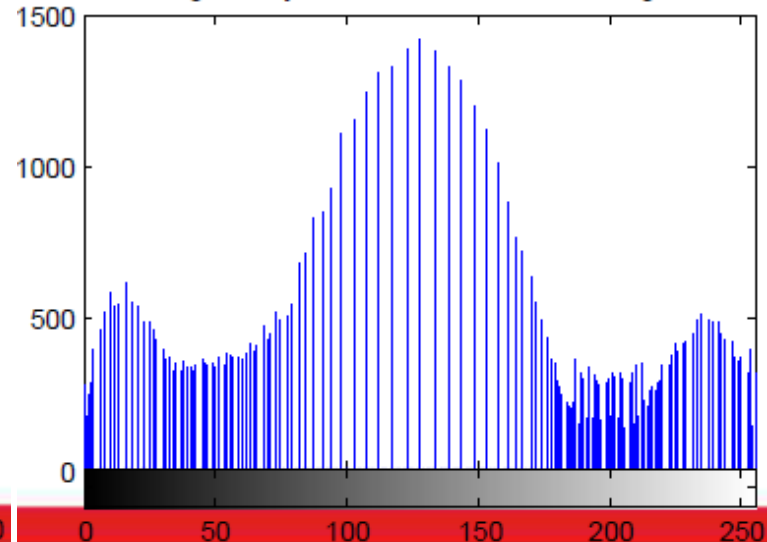
Histogram Eşitleme Sonrası Resim



Orjinal Resim Histogramı



Histogram Eşitleme Sonrası Resim Histogramı



# Ödev

- Gri bir görüntü üzerinde histogram eşitleme ve histogram germe işlemlerini gerçekleştiren MATLAB kodlarını yazınız.
- Not: Hazır Matlab fonksiyonu kullanılmayacaktır.
- Teslim tarihi: 04 Nisan 2019

