



# BSM 409

# GÖRÜNTÜ İŞLEME

Hafta - 5

Histogram – Kontrast – Parlaklık – Yoğunluk Dönüşüm İşlemleri

Doç. Dr. Eftal ŞEHİRLİ

---

Bilgisayar Mühendisliği Bölümü

# İçerik

- Histogram
- Kontrast
- Parlaklık
- Dönüşümler

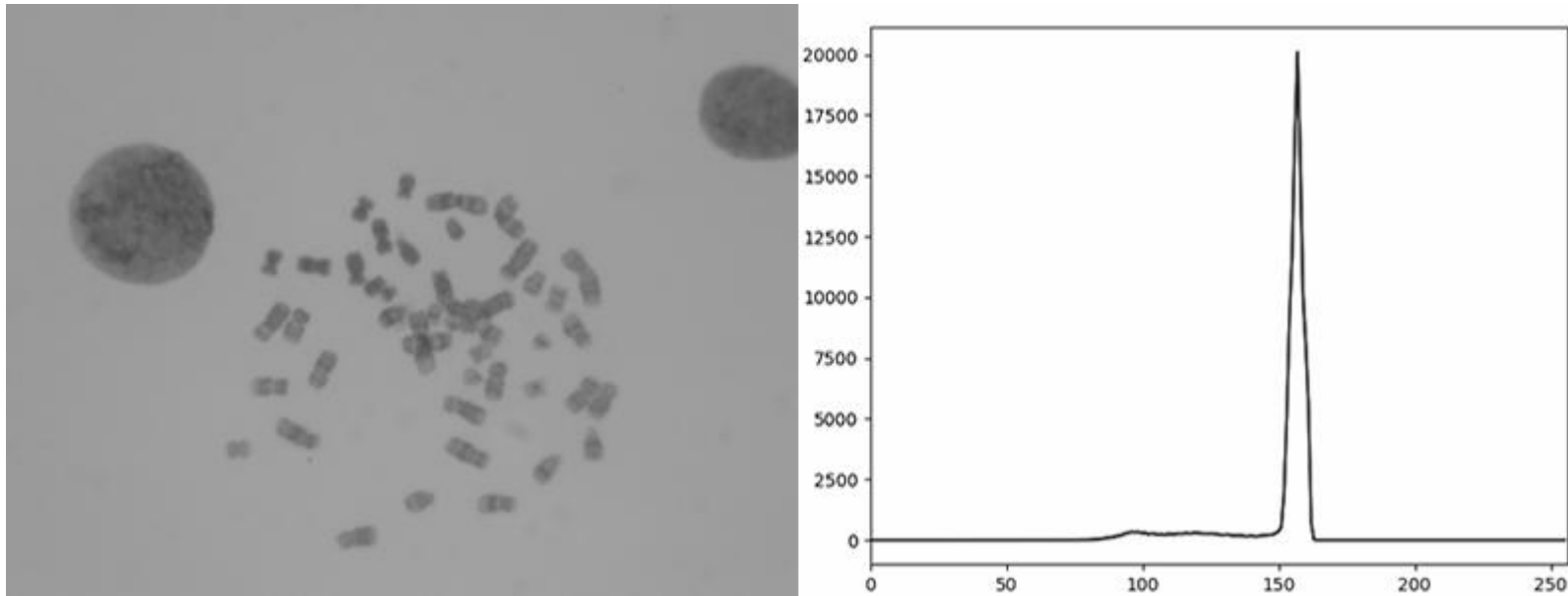


# Tanım

- Histogram: Her piksel yoğunluk değerinin görüntüdeki miktarını (frekansını) gösterir.
- Kontrast: Görüntülerde karanlık ile aydınlık alan arasındaki fark ya da oran, zıtlık
- Parlaklık: Işık ya da aydınlık değeri

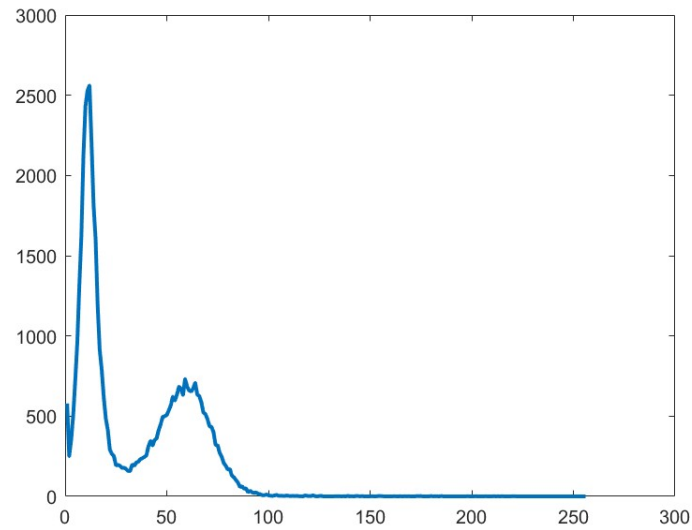
# Histogram

- Her piksel yoğunluk değerinin görüntüdeki miktarını (frekansını) gösterir.
- Yoğunluk değeri 0-255 yoğunluk değerlerinin frekansları nedir?

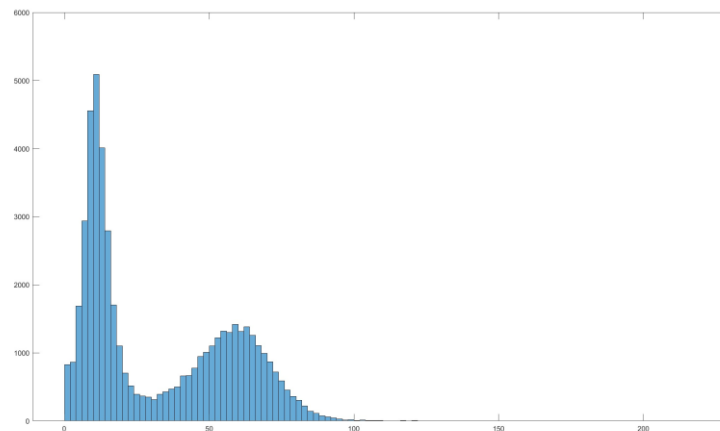


Beyaz Kan Hücresi (WBC) Görüntüsü ve Histogram Grafiği

# Histogram – Örnek Görüntü



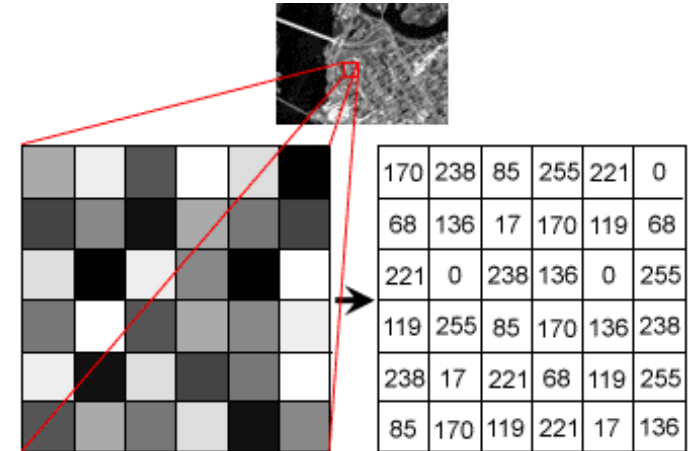
Beyin MR Görüntüsü  
Programı Yazılmış Histogram Grafiği



Beyin MR Görüntüsü  
Matlab Histogram Grafiği

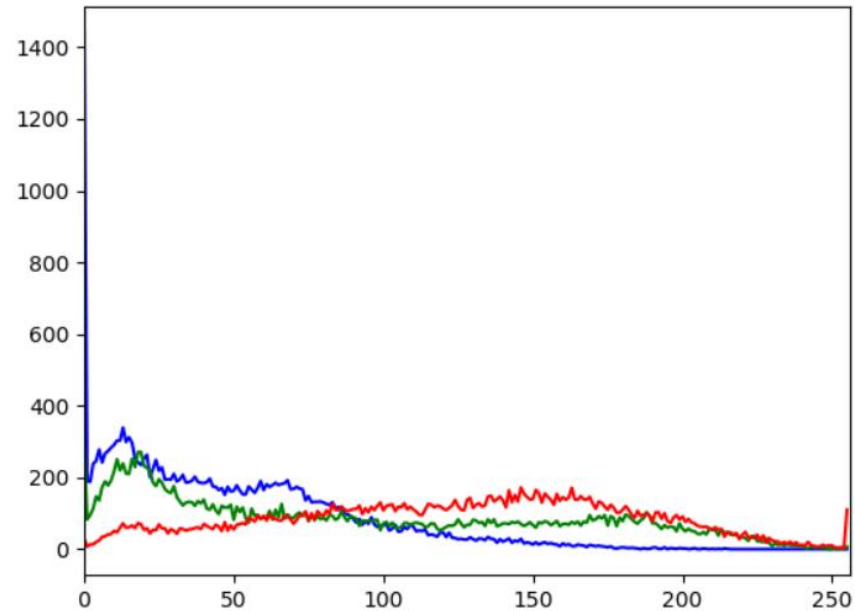
# Histogramın Elde Edilmesi

- Histogramlar birçok uzamsal görüntü işleme tekniğinin temelini oluşturmaktadır. Görüntünün zenginleştirilmesi başta olmak üzere görüntü sıkıştırma, bölütleme gibi birçok görüntü işleme aşamasında kullanılabilir.
- Temelde istatistiğe dayanan bu yöntem, görüntüyü görmeden sadece histogramına bakarak görüntünün durumu hakkında (koyu, açık, düşük kontrastlı, yüksek kontrastlı vs.) yorum yapmayı sağlayabilmektedir.
- Histogramlar herhangi boyuttaki veriler için kolayca hesaplanabilir. Basit ve düşük donanımlı sistemlerde kullanışlı olması, gerçek zamanlı görüntülerin işlenmesinde performans sağlaması yöntemi daha popüler hale getirmektedir.
- Görüntünün histogramını elde edebilmek için her intensity düzeyi için bir vektör veya liste tanımlanır ve başlangıç değeri sıfır olarak belirlenir. Daha sonra görüntüdeki tüm pikseller gezilerek pikselin intensity değerine göre vektörün değeri artırılır. Böylece, histogram elde edilmiş olur.



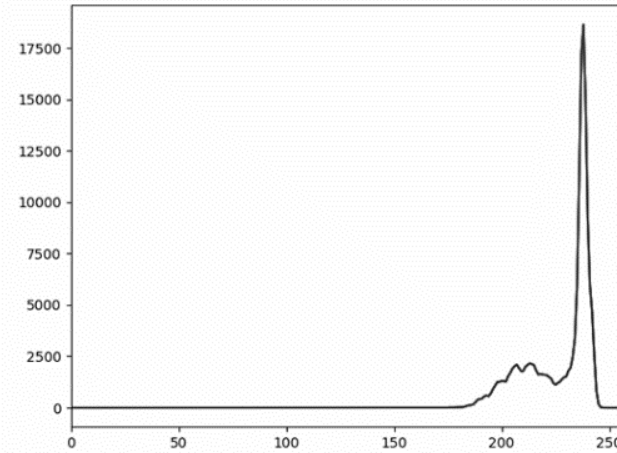
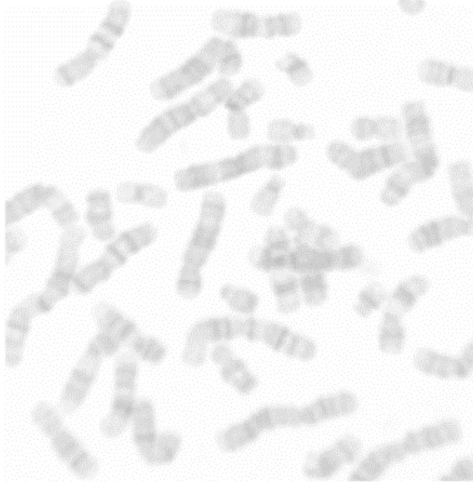
# Histogramın Elde Edilmesi

- Histogram tek bir renk kanalı üzerinde çıkarılabileceği gibi R, G ve B olmak üzere üç ayrı kanal içinde çıkarılarak incelenebilir. Şekilde görüntüye ait üç farklı kanalda histogramları verilmiştir.

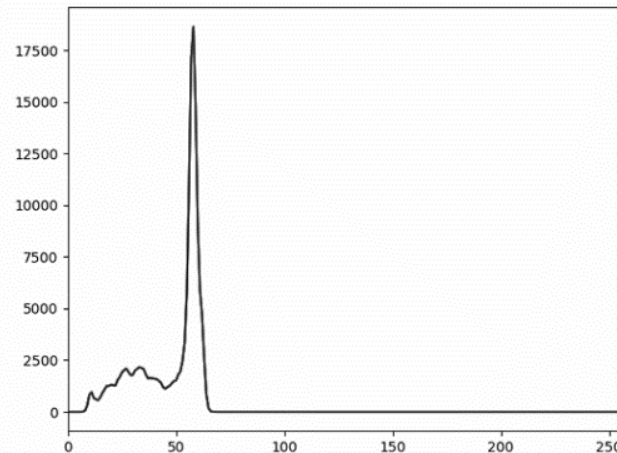
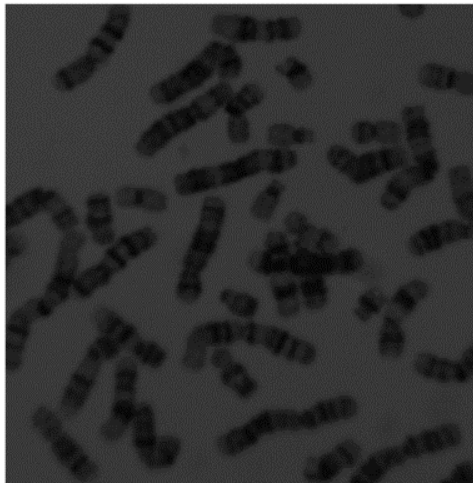


# Histogram – Koyu vs Açık Renkli Görüntü

açık görüntü ve histogramı



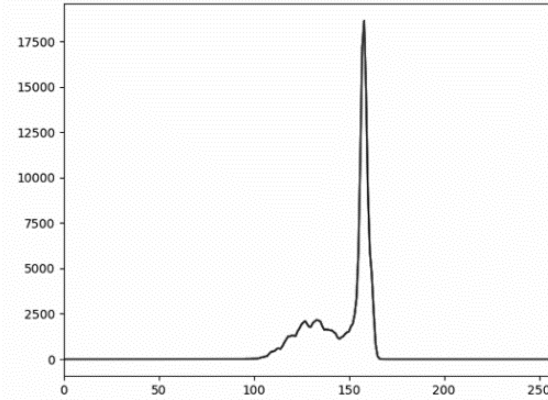
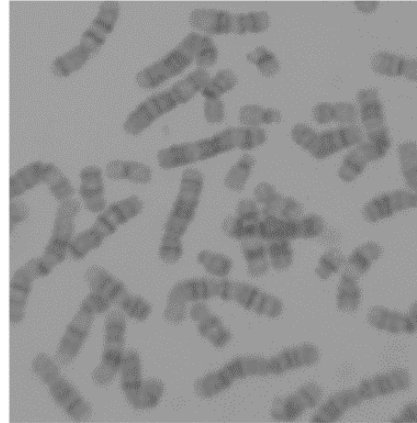
koyu görüntü ve histogramı



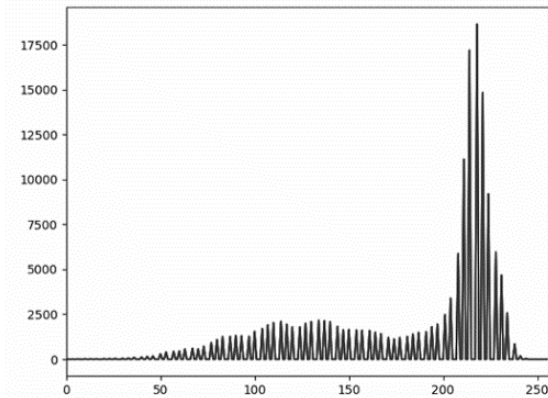
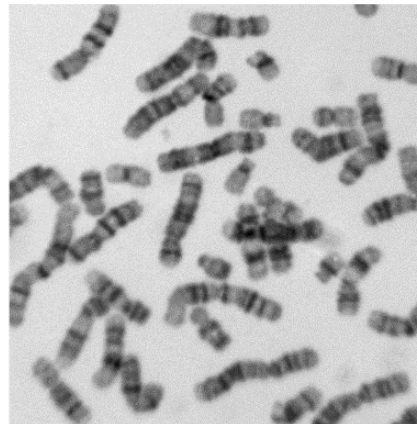


# Histogram – Yüksek vs Düşük Kontrastlı Görüntü

düşük kontrastlı görüntü  
ve histogramı



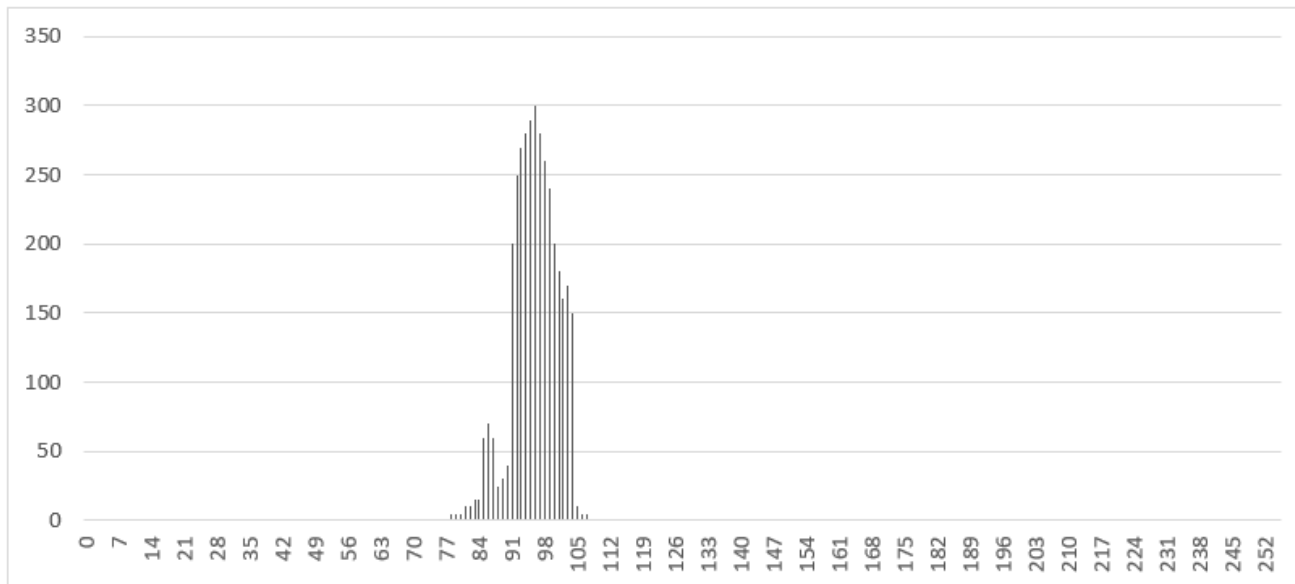
yüksek kontrastlı görüntü  
ve histogramı



# Histogram (Kontrast) Germe (Contrast Streching)

- Dar bir alana toplanmış histogram değerlerini dağıtarak detayları belirginleştirmek için kullanılır.

$$P_{çıktı} = (P_{girdi} - c) \frac{g_{max} - g_{min}}{d - c} + g_{min}$$



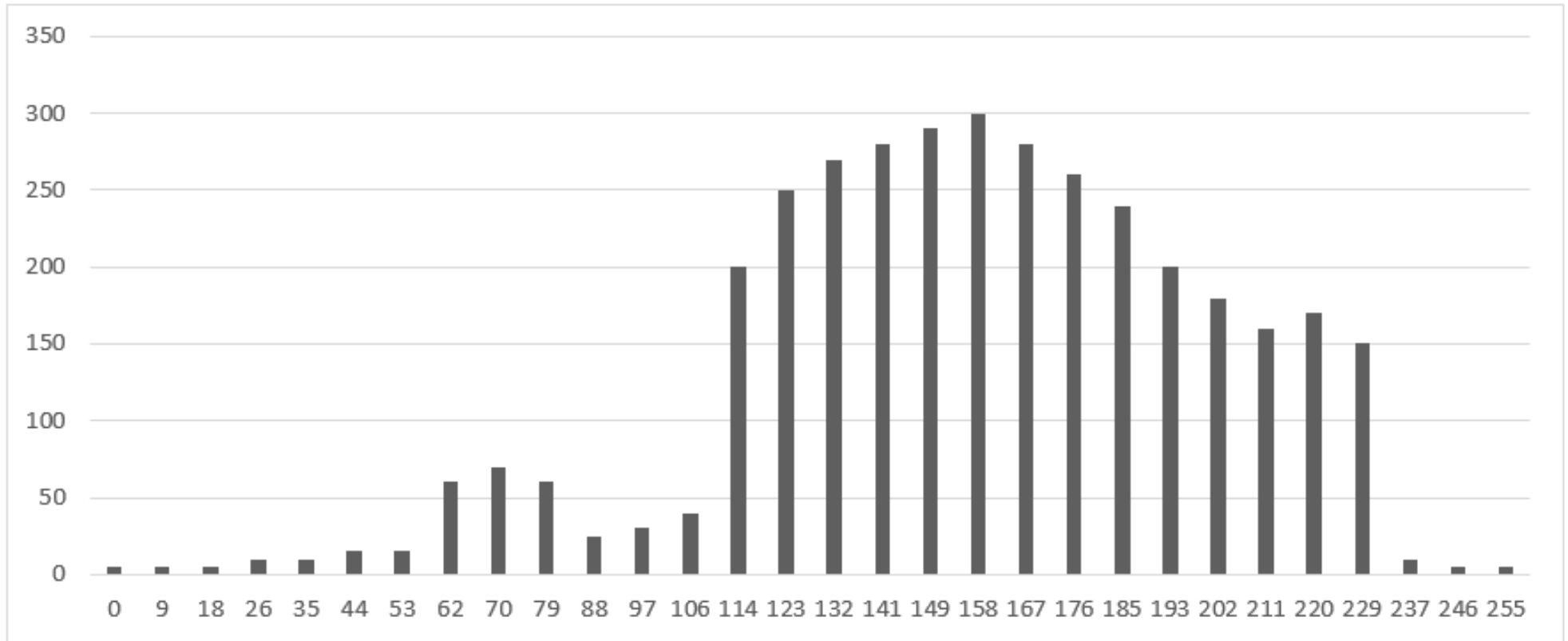
## Örnek - Görüntü Piksel Yoğunluk Değerleri

$$P_{çıkı} = (P_{girdi} - c) \frac{g_{max} - g_{min}}{d - c} + g_{min}$$

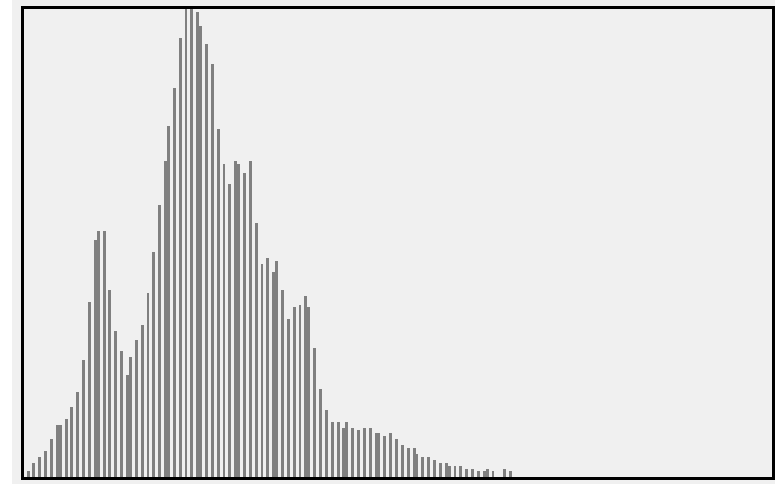
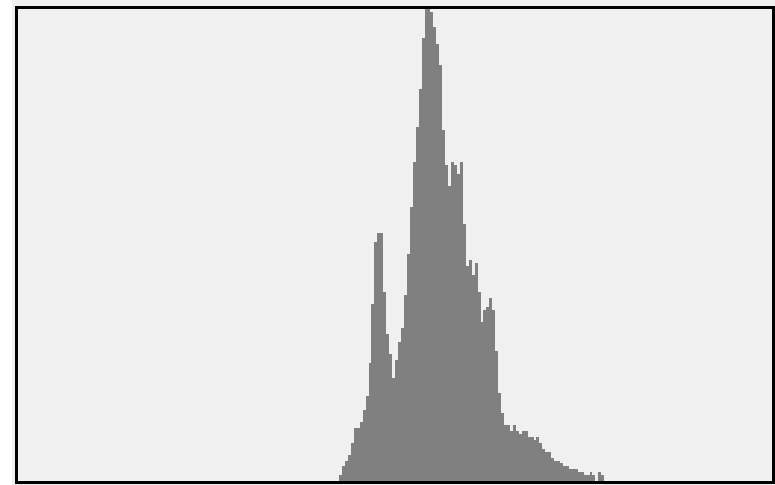
$P_{girdi}$	$P_{çıkı}$	$P_{girdi}$	$P_{çıkı}$	$P_{girdi}$	$P_{çıkı}$
78	0	88	88	98	176
79	9	89	97	99	185
80	18	90	106	100	193
81	26	91	114	101	202
82	35	92	123	102	211
83	44	93	132	103	220
84	53	94	141	104	229
85	62	95	149	105	237
86	70	96	158	106	246
87	79	97	167	107	255

$g_{max}=255$   
 $g_{min}=0$   
 $d=107$   
 $c=78$

# Histogram (Kontrast) Germe - Sonuç

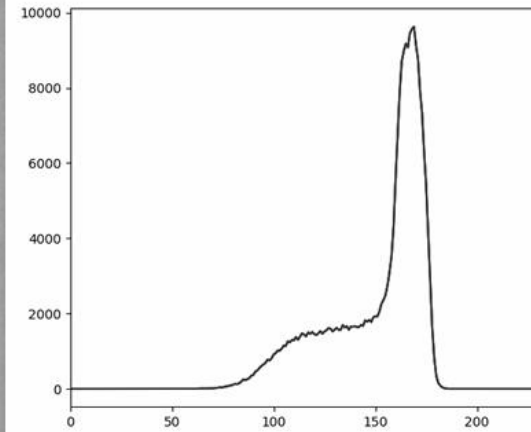


# Histogram (Kontrast) Germe - Örnek



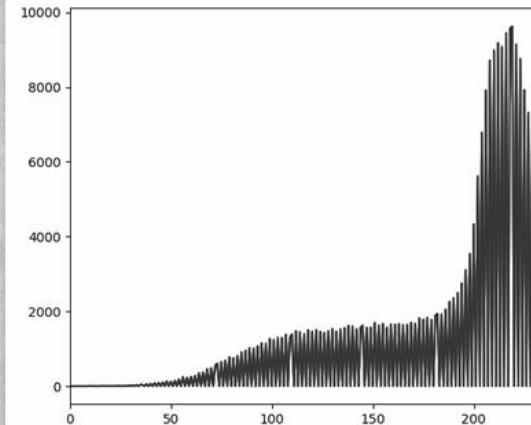
# Histogram (Kontrast) Germe - Örnek

amecV1[3] kullanılarak yapılan tarama sonucu elde edilen görüntülerden toplamda 3018 tane nesne seçilmiş ve uzman değerlendirilmesine tabi tutulmuştur. Bu görüntüler içinden oluşturulan filtre sabit açılı yöntemi kullanarak 184 görüntüyü metafaz olarak belirlemiş, yapılan uzman incelemesi sonucunda ise işaretlenen 29 görüntünün metafaz olmadığı belirlenmiştir. İçerik tabanlı benzerlik yöntemi kullanılarak yapılan taramada ise 184 görüntü metafaz olarak belirlenmiştir. Uzman değerlendirmesi sonucunda ise bu görüntülerden 22 tanesinin metafaz olmadığı tespit edilmiştir.



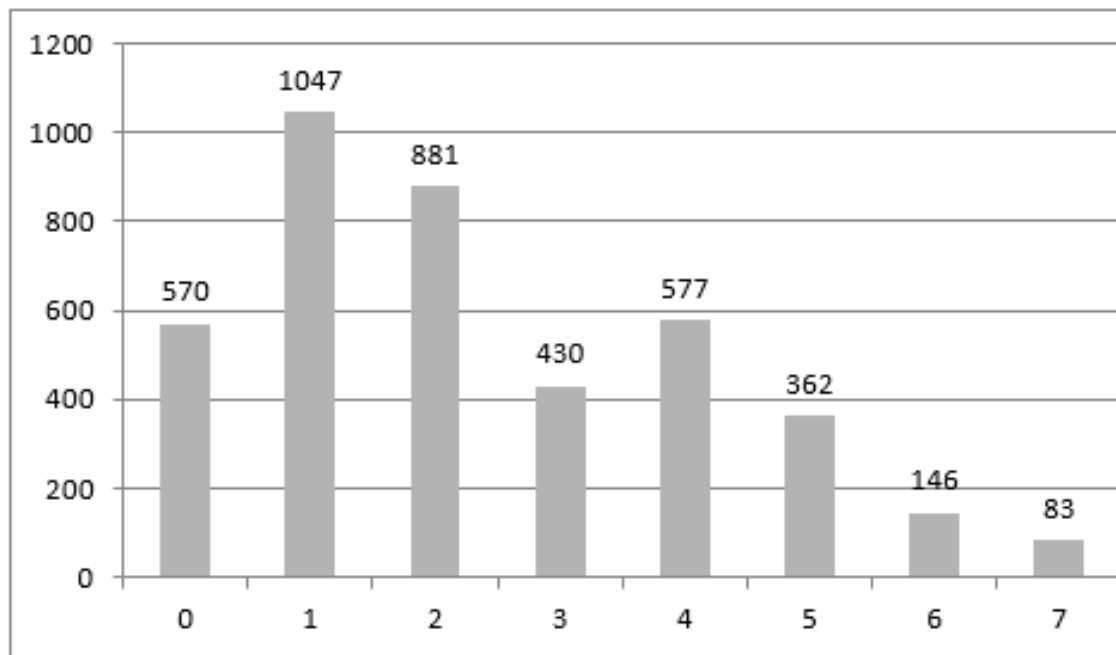
a) kontrastı kötü görüntü ve histogramı

amecV1[3] kullanılarak yapılan tarama sonucu elde edilen görüntülerden toplamda 3018 tane nesne seçilmiş ve uzman değerlendirilmesine tabi tutulmuştur. Bu görüntüler içinden oluşturulan filtre sabit açılı yöntemi kullanarak 184 görüntüyü metafaz olarak belirlemiş, yapılan uzman incelemesi sonucunda ise işaretlenen 29 görüntünün metafaz olmadığı belirlenmiştir. İçerik tabanlı benzerlik yöntemi kullanılarak yapılan taramada ise 184 görüntü metafaz olarak belirlenmiştir. Uzman değerlendirmesi sonucunda ise bu görüntülerden 22 tanesinin metafaz olmadığı tespit edilmiştir.



b) histogram germe sonucu oluşan görüntü ve histogramı

# Histogram Eşitleme (Histogram Equalization)



$$T(k) = L - 1 \sum_{n=0}^k p_n$$

- Görüntüde kontrast zenginleştirilmesi uygulanır.
- Histogramı verilen 3 bit gri değer aralıklı 64x64lık bir görüntü
- Not: 4096 pikseli bir görüntü

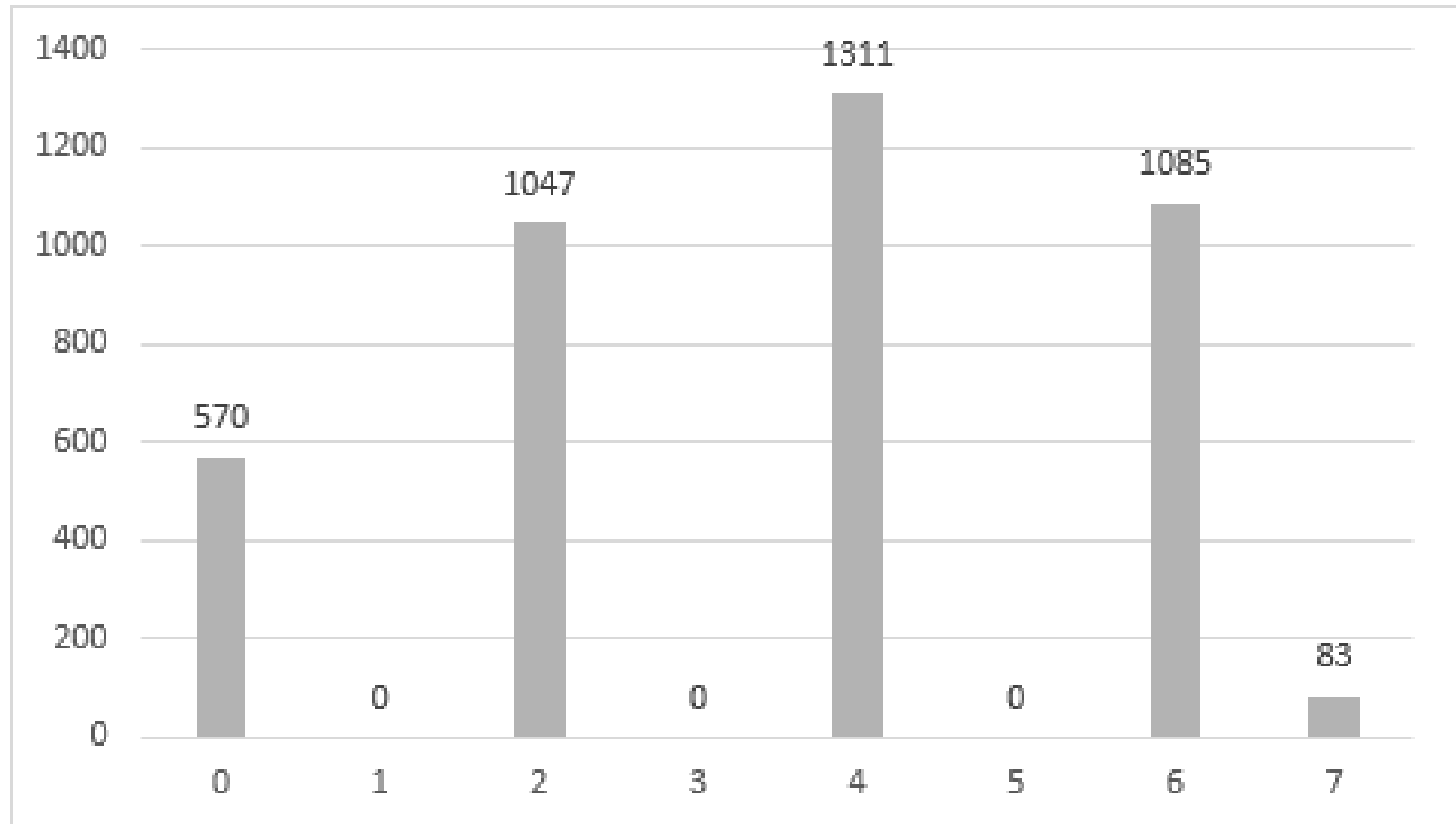
# Histogram Eşitleme - Örnek

$$T(k) = L - 1 \sum_{n=0}^k p_n$$

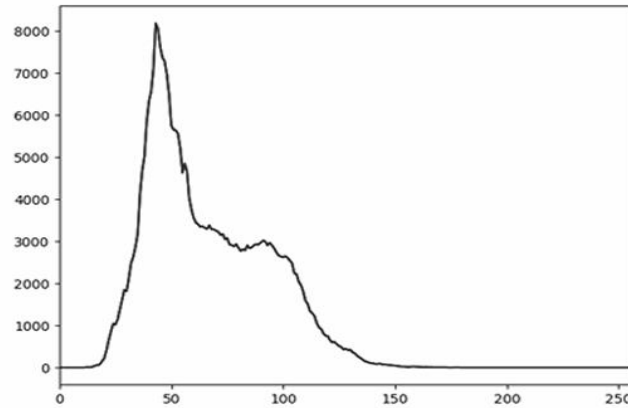
$r_k$	$n_k$	$p_n$	$Kn_k$	$Kn_k$	$Kp_n$	$T(k)$	$\text{floor}(T(k))$
0	570	0,14	570	570	0,14	0,98	0
1	1047	0,26	570+1047	1617	0,39	2,73	2
2	881	0,22	570+1047+881	2498	0,61	4,27	4
3	430	0,1	570+1047+881+430	2928	0,71	4,97	4
4	577	0,14	570+1047+881+430+577	3505	0,86	6,02	6
5	362	0,09	570+1047+881+430+577+362	3867	0,94	6,58	6
6	146	0,04	570+1047+881+430+577+362+146	4013	0,98	6,86	6
7	83	0,02	570+1047+881+430+577+362+146+83	4096	1	7	7



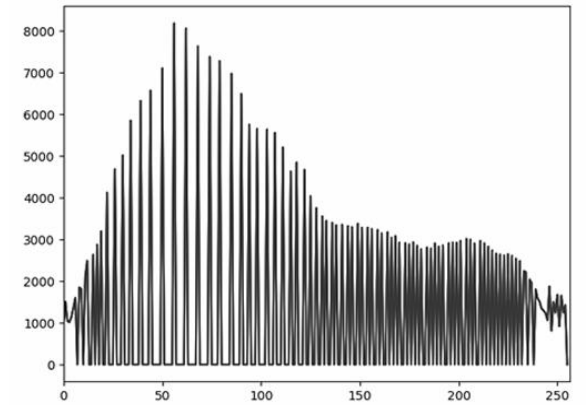
# Histogram Eşitleme - Örnek



# Örnek – Histogram Eşitleme

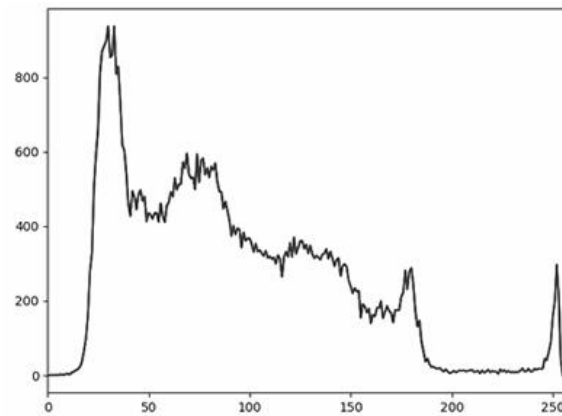


a) orijinal görüntü ve histogramı

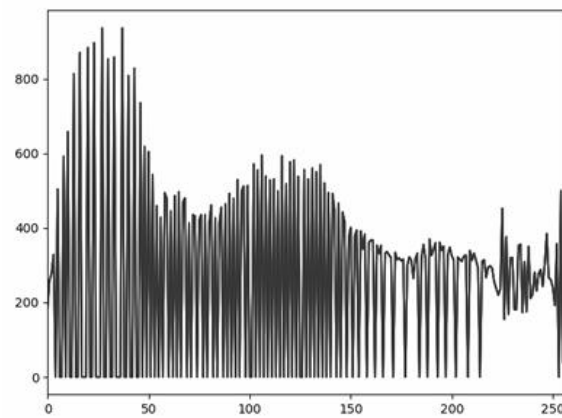


b) histogramı eşitlenmiş görüntü ve histogramı

# Örnek – Histogram Eşitleme



a) orijinal görüntü ve histogramı



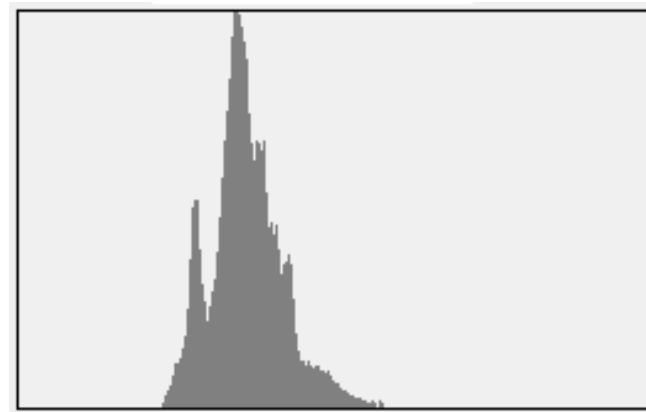
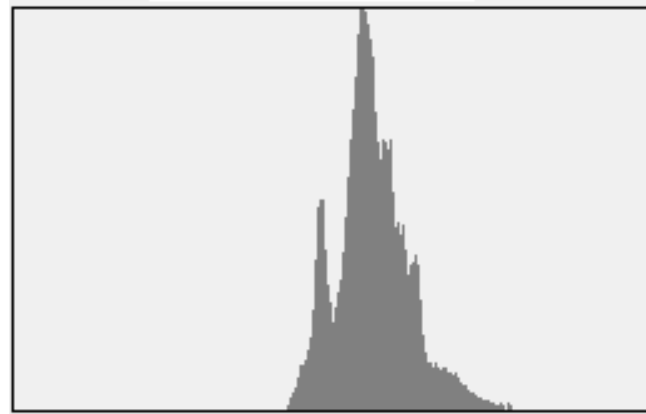
b) histogramı eşitlenmiş görüntü ve histogramı

## Parlaklık (Histogram Kaydırma)(Brightness)

- Kırmızı, yeşil ve mavi renk değerlerine belirtildiği kadar parlaklık değeri eklenir veya çıkarılır.
- Elde edilen yeni değer 0-255 arasında değilse, bu değerlere indirgenir veya yükseltilir.

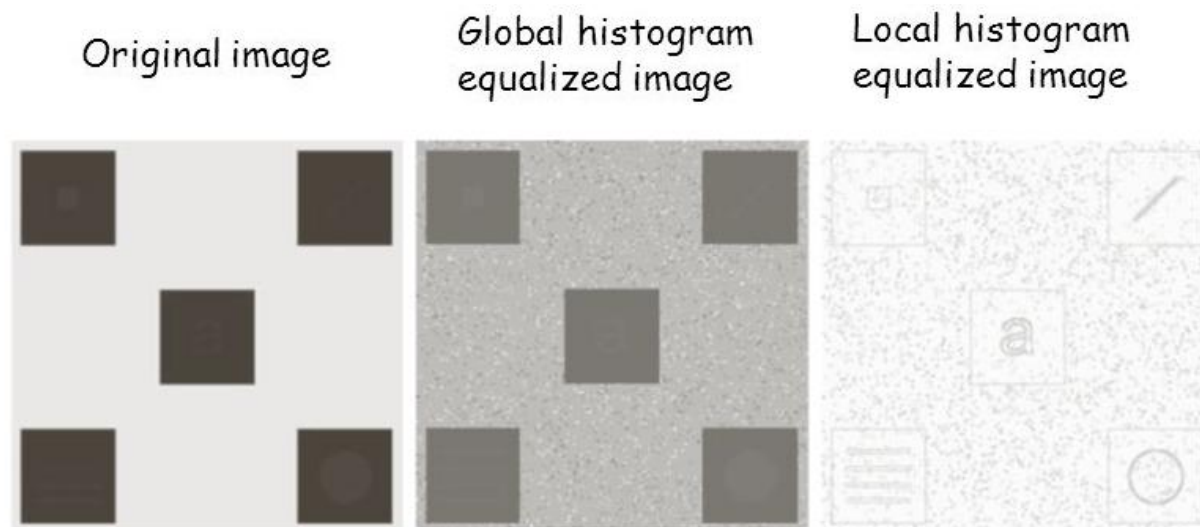


# Parlaklık – Histogram Kaydırma




# Lokal Histogram İşleme

- Bölgesel histogram germe,
- Bölgesel histogram eşitleme,
- Bölgesel histogram kaydırma,

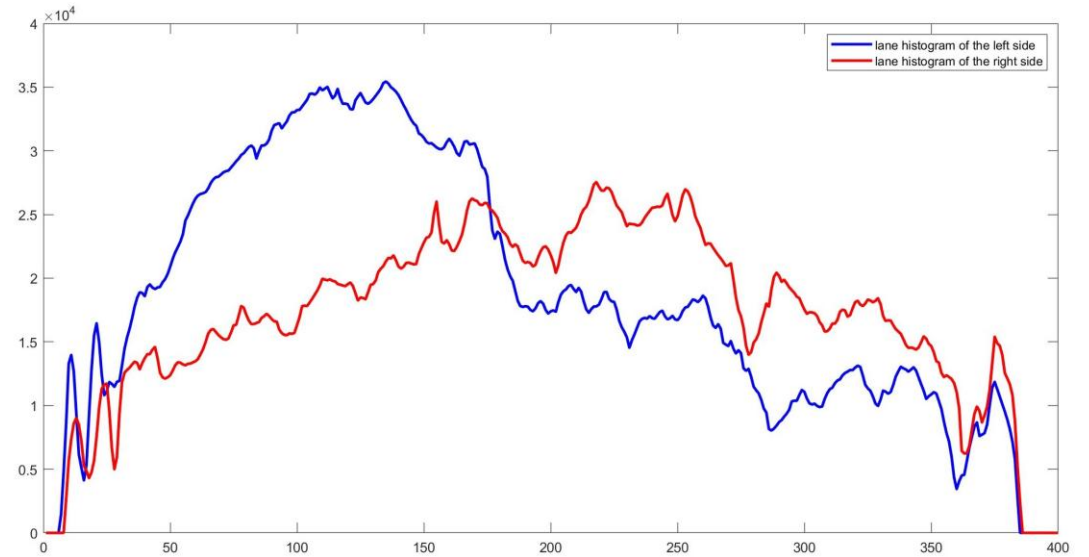
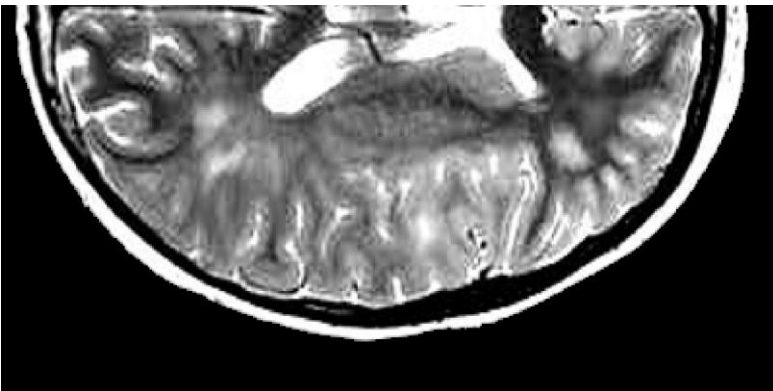
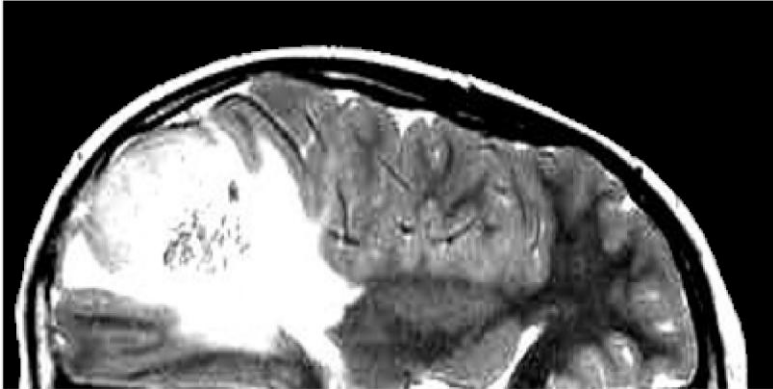


# Piksel Profil Analizi – Şerit Histogram (Pixel Profile Analysis – Lane Histogram)

- Piksel yoğunluk değerlerinin dikey ya da yatay olarak toplanması ile elde edilir.

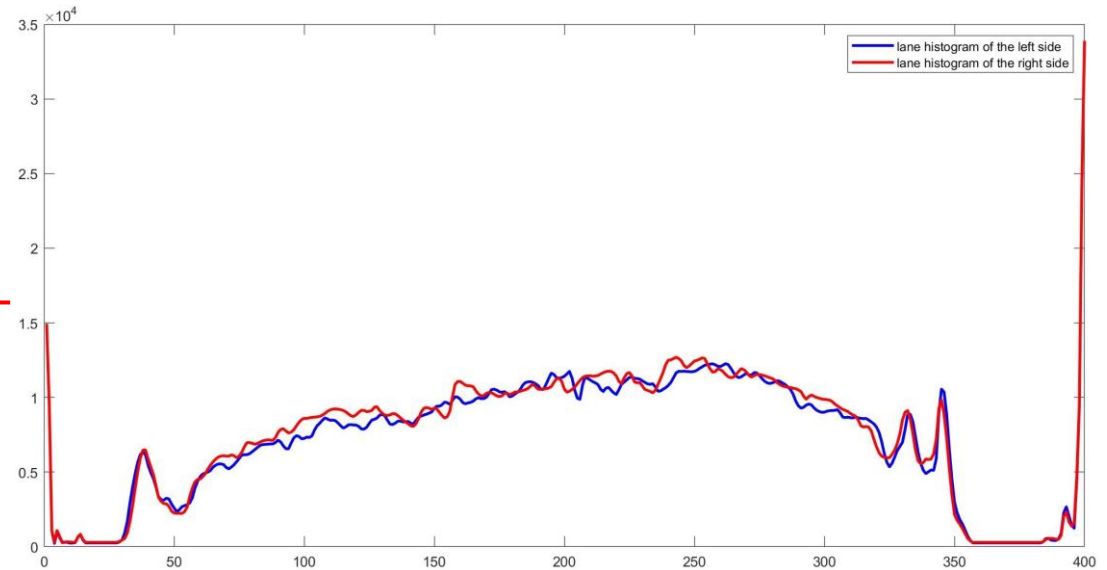
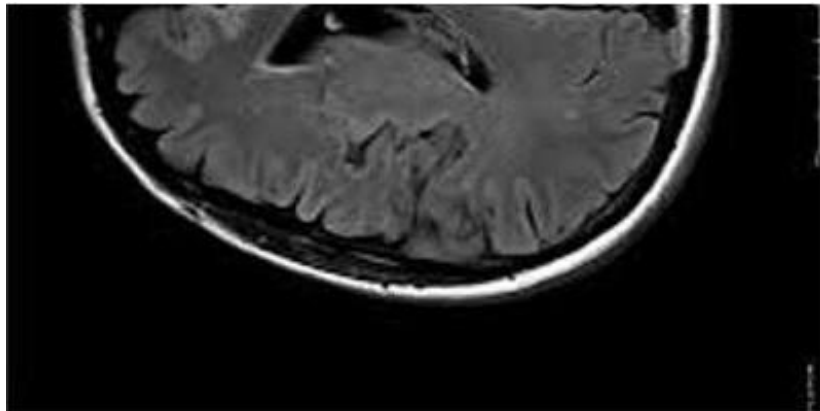
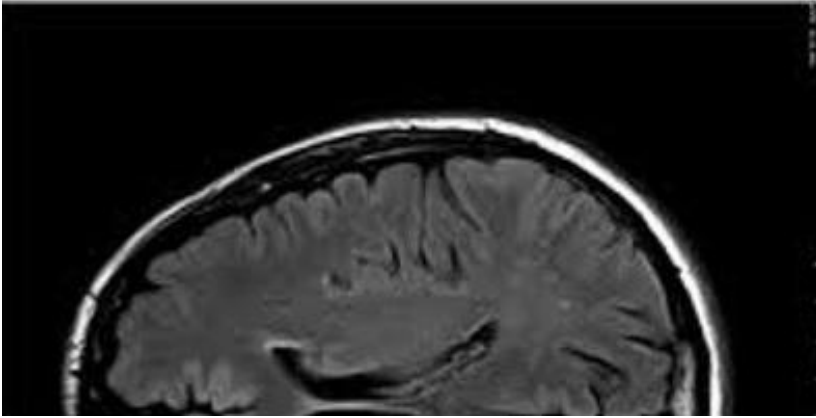
5	12	17	14	7	4
6	11	16	13	7	5
7	11	14	12	8	6
8	10	11	10	9	8
10	9	8	8	10	10
11	8	5	6	10	12
13	7	3	5	11	13
13	7	2	4	11	14
6	6	7	9	10	11
0	1	4	7	11	15
					
79	82	87	88	94	98

# Örnek – Şerit Histogram – Tümörlü Beyin Görüntüsü

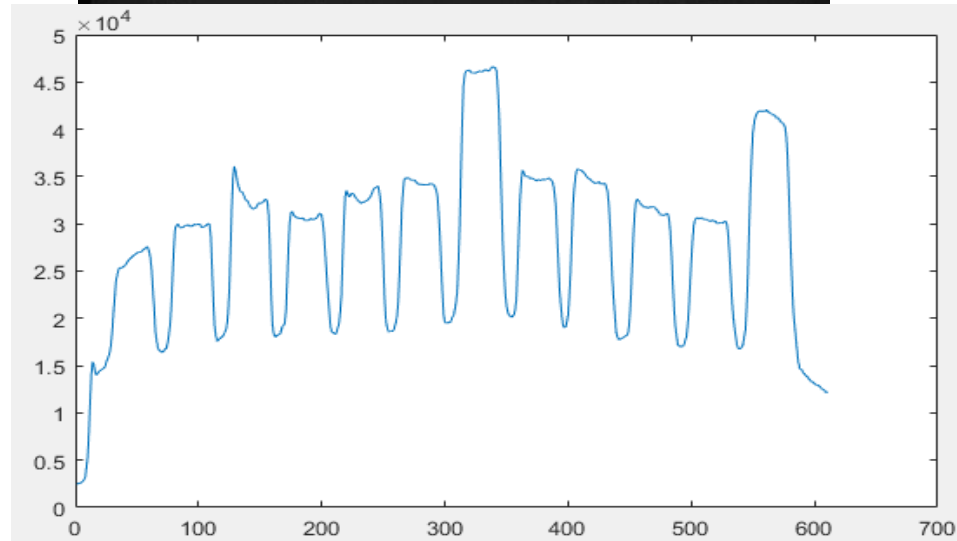
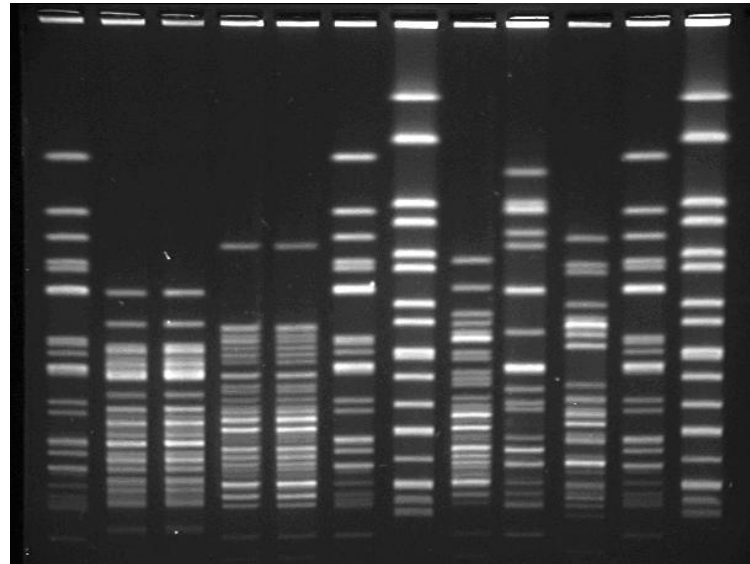




# Örnek – Şerit Histogram – Tümörsüz Beyin Görüntüsü

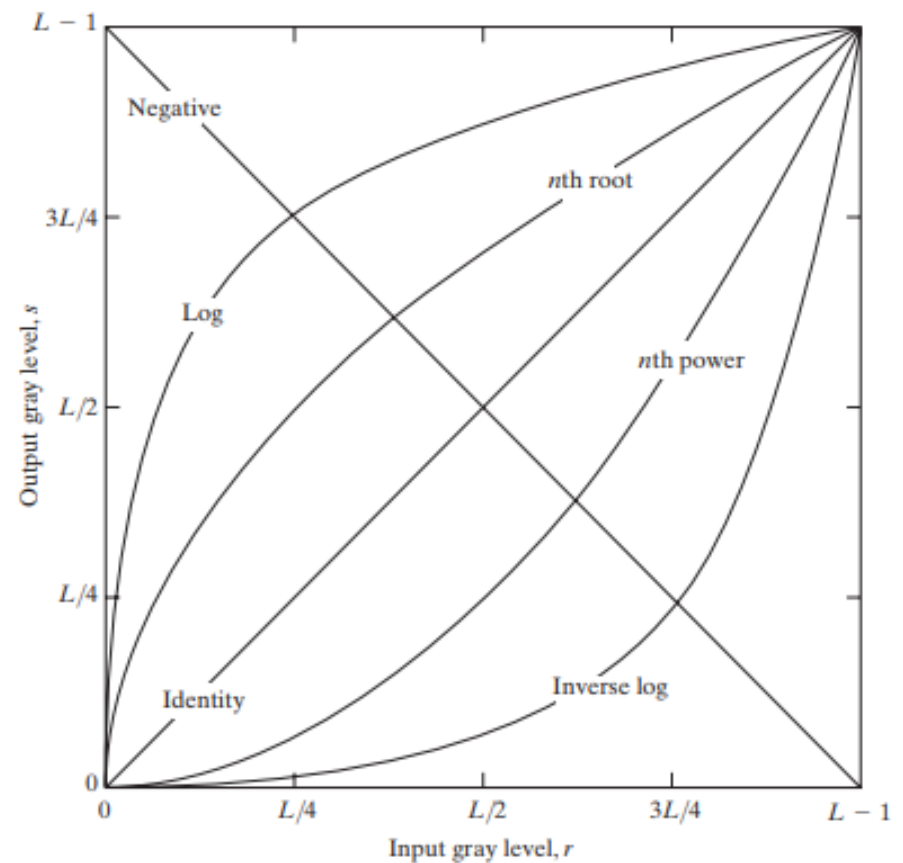


# Örnek – Şerit Histogram – DNA Jel Elektroforez Görüntüsü



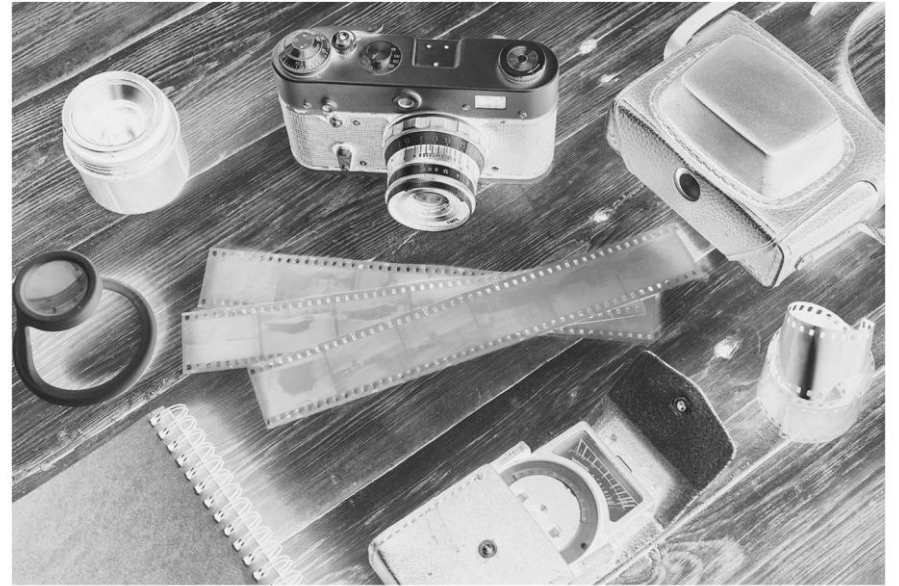
# Temel Bazı Yoğunluk Dönüşüm İşlemleri

- Image Negatives (Linear)
- Log Transformations
- Power-Law (Gamma) Transformations
- Piecewise-Linear Transformation Functions



# Negatif Dönüşüm

$$s = L - 1 - r$$



# Log Dönüşümü

- Pratik olarak, log dönüşümü dar bir aralıktaki düşük yoğunluklu piksel yoğunluk değerlerini geniş bir aralıktaki piksel yoğunluk değerlerine eşler.

- Matematikteki log dönüşüm formülü:

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

$$r \geq 0,$$

$r$  görüntü piksel yoğunluk değeri (log dönüşüm öncesi)

$s$  görüntü piksel yoğunluk değeri (log dönüşüm sonrası)

$c$  sabit değer

$$c = 255/(\log(1+m))$$

$m$  görüntüdeki maksimum yoğunluk değeri

# Log Dönüşümü

```
import cv2
import numpy as np

# Open the image.
img = cv2.imread('sample.jpg')

# Apply log transform.
c = 255/(np.log(1 + np.max(img)))
log_transformed = c * np.log(1 + img)

# Specify the data type.
log_transformed = np.array(log_transformed, dtype = np.uint8)

# Save the output.
cv2.imwrite('log_transformed.jpg', log_transformed)
```





# Gama Dönüşümü (Power-Law Transformation)

- Gama düzeltmesi, farklı ekran ayarlarına sahip farklı türdeki monitörlerden görüntüler izlendiğinde görüntülerin solmasını veya koyulaşmasını önlemek için görüntüleri ekranda doğru şekilde görüntülemek açısından önemlidir. Bunun nedeni, gözlerimizin görüntüleri gama eğrisi şeklinde algılaması, kameraların ise görüntüleri doğrusal bir şekilde yakalamasıdır.
- Matematikteki formülü:
- $s=cr^\gamma$

$\gamma$  gama değeri



# Gama Dönüşümü

```
import cv2
import numpy as np

# Open the image.
img = cv2.imread('sample.jpg')

# Trying 4 gamma values.
for gamma in [0.1, 0.5, 1.2, 2.2]:

    # Apply gamma correction.
    gamma_corrected = np.array(255*(img / 255) ** gamma, dtype = 'uint8')

    # Save edited images.
    cv2.imwrite('gamma_transformed'+str(gamma)+'.jpg', gamma_corrected)
```



# Gama Dönüşümü (Power-Law Transformation)

$\gamma = 0.1$



$\gamma = 0.5$



$\gamma = 1.2$

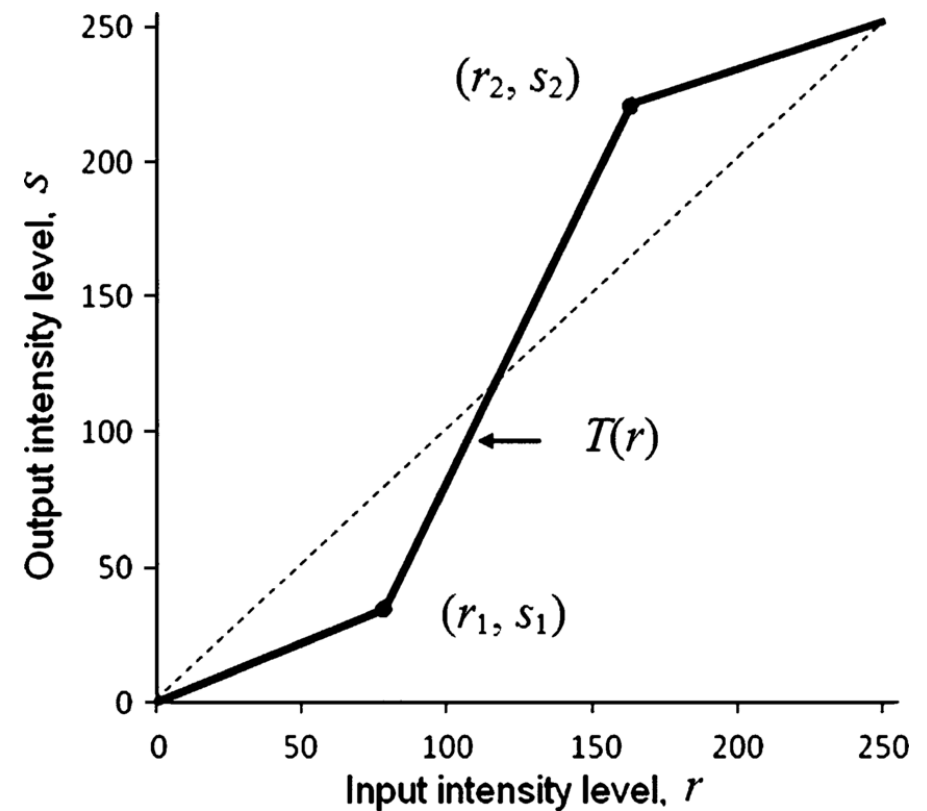


$\gamma = 2.2$



# Parçalı Doğrusal Dönüşüm (Piecewise - Linear Transformation)

- Yoğunluk değerlerini (piksel değerlerini) **parçalı doğrusal fonksiyonlarla** başka değerlere dönüştürür.
- Tek bir doğrusal denklem yerine, yoğunluk eksenini birkaç **aralık**'a bölünür ve her aralıkta farklı bir **doğrusal fonksiyon** uygulanır.
- Özellikle **kontrast iyileştirme** ve **parlaklık ayarlama** için tercih edilir.





# Parçalı Doğrusal Dönüşüm (Piecewise - Linear Transformation)

```
import cv2
import numpy as np

# Function to map each intensity level to output
# intensity level.
def pixelVal(pix, r1, s1, r2, s2):
    if (0 <= pix and pix <= r1):
        return (s1 / r1)*pix
    elif (r1 < pix and pix <= r2):
        return ((s2 - s1)/(r2 - r1)) * (pix - r1) + s1
    else:
        return ((255 - s2)/(255 - r2)) * (pix - r2) + s2

# Open the image.
img = cv2.imread('sample.jpg')

# Define parameters.
r1 = 70
s1 = 0
r2 = 140
s2 = 255

# Vectorize the function to apply it to each value in the
# Numpy array.
pixelVal_vec = np.vectorize(pixelVal)

# Apply contrast stretching.
contrast_stretched = pixelVal_vec(img, r1, s1, r2, s2)

# Save edited image.
cv2.imwrite('contrast_stretch.jpg', contrast_stretched)
```

# Parçalı Doğrusal Dönüşüm (Piecewise - Linear Transformation)





- Ders sonu