

# BSM409 Görüntü İşleme

## Bölüm 3 Görüntü İşleme ile İlgili Temel Kavramlar

Dr. Öğr. Üyesi Caner ÖZCAN

Those who wish to succeed must ask the right preliminary questions.  
(Başarmak isteyenler doğru başlangıç soruları sormalıdır.) ~Aristotle

# İçerik

## 2. Sayısal Görüntü Temelleri

- ▶ Görsel Algının Unsurları
- ▶ Işık ve Elektromanyetik Spektrum
- ▶ Görüntü Algılama ve Edinme
- ▶ Görüntü Örnekleme ve Nicemleme
- ▶ Pikseller Arasındaki Bazı Temel İlişkiler
- ▶ Sayısal Görüntü İşlemede Kullanılan Matematiksel İşlemlere Giriş
- ▶ Python Görüntü İşleme

# Pikseller Arasındaki Bazı Temel İlişkiler

- ▶  $(x,y)$  koordinatındaki bir  $p$  pikseli **komşuları**
- ▶  $p$ 'nin 4-komşusu için  **$N_4(p)$**  olarak:  
 $(x-1, y), (x+1, y), (x,y-1)$  ve  $(x, y+1)$ .
- ▶  $p$ 'nin 4 köşegen komşusu için  **$N_D(p)$**  olarak:  
 $(x-1, y-1), (x+1, y+1), (x+1,y-1)$  ve  $(x-1, y+1)$ .
- ▶  $p$ 'nin 8-komşusu için  **$N_8(p)$**  olarak  
$$N_8(p) = N_4(p) \cup N_D(p)$$

# Pikseller Arasındaki Bazı Temel İlişkiler

## ► Bitişiklik

$V$  bitişikliği tanımlamak için kullanılan yeğinlik değerleri kümesi olsun.

İkili bir görüntüde, eğer 1 değerli piksellerin bitişikliğinden bahsedersek  $V=\{1\}$ 'dir.

► **4-bitişiklik:** Eğer  $q \in N_4(p)$  kümesinde ise, değerleri  $V$ 'den olan  $p$  ve  $q$  pikselleri 4-bitişiktir.

► **8-bitişiklik:** Eğer  $q \in N_8(p)$  kümesinde ise, değerleri  $V$ 'den olan  $p$  ve  $q$  pikselleri 8-bitişiktir.

# Pikseller Arasındaki Bazı Temel İlişkiler

## ► Bitişiklik

$V$  bitişikliği tanımlamak için kullanılan yeğinlik değerleri kümesi olsun.

Gri seviyesindeki piksellerin bitişikliğinde  $V$  kümesi bu 256 değer herhang bir alt kümesidir.

► **m-bitişiklik:** Değerleri  $V'$ 'den olan  $p$  ve  $q$  pikselleri  $m$ -bitişiktir, eğer

(i)  $q \in N_4(p)$  kümesinde ise, veya

(ii)  $q \in N_D(p)$  kümesinde ve  $N_4(p) \cap N_4(q)$  kümesinin piksel değerleri  $V'$ 'de değilse.

# Pikseller Arasındaki Bazı Temel İlişkiler

## Yol

- ▶  $(x,y)$  koordinatlı  $p$  pikselinden  $(s,t)$  koordinatlı  $q$  pikseline bir (sayısal) yol (veya eğri)  
 $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$   
koordinatlarına sahip farklı piksellerin oluşturduğu bir dizidir.  
  
Burada,  $(x_i, y_i)$  ve  $(x_{i-1}, y_{i-1})$  pikselleri  $1 \leq i \leq n$  için bitişiktir.
- ▶ Bu durumda  $n$  yolun uzunluğudur.
- ▶ Şayet  $(x_0, y_0) = (x_n, y_n)$  ise yol **kapalı** yoldur.
- ▶ Belirtilen bitişikliğin tipine bağlı olarak 4-, 8- veya m-yollar tanımlayabiliriz.

# Örnekler: Bitişiklik ve Yol

$$V = \{1\}$$

0 1 1

0 1 0

0 0 1

0 1 1

0 1 0

0 0 1

0 1 1

0 1 0

0 0 1

# Örnekler: Bitişiklik ve Yol

$$V = \{1\}$$

0 1 1

0 1 0

0 0 1

0 1 1

0 1 0

0 0 1

8-bitişik

0 1 1

0 1 0

0 0 1



# Örnekler: Bitişiklik ve Yol

$$V = \{1\}$$

0	1	1
0	1	0
0	0	1

8-bitişik

0	1	1
0	1	0
0	0	1

m-bitişik

0	1	1
0	1	0
0	0	1

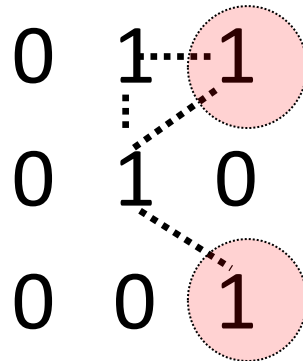
# Örnekler: Bitişiklik ve Yol

$$V = \{1\}$$

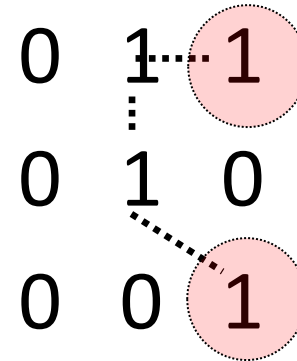
$0_{1,1}$   $1_{1,2}$   $1_{1,3}$

$0_{2,1}$   $1_{2,2}$   $0_{2,3}$

$0_{3,1}$   $0_{3,2}$   $1_{3,3}$



**8-bitişik**



**m-bitişik**

(1,3)'den (3,3)'e 8-yol:

(i) (1,3), (1,2), (2,2), (3,3)

(ii) (1,3), (2,2), (3,3)

(1,3)'den (3,3)'e m-yol:

(1,3), (1,2), (2,2), (3,3)

# Uzaklık Ölçütleri

► Sırasıyla koordinatları  $(x, y)$ ,  $(s, t)$  ve  $(v, w)$  olan  $p$ ,  $q$  ve  $z$  pikselleri için  $D$ , bir uzaklık fonksiyonu veya bir metriktir eğer:

- $D(p, q) \geq 0$     [ $D(p, q) = 0$ , ancak ve ancak  $p = q$ ]
- $D(p, q) = D(q, p)$  ve
- $D(p, z) \leq D(p, q) + D(q, z)$  ise.

# Uzaklık Ölçütleri

► Aşağıdakiler farklı uzaklık ölçütleridir.

a. Öklid Uzaklığı:

$$D_e(p, q) = [(x-s)^2 + (y-t)^2]^{1/2}$$

b. Şehir-Blok Uzaklığı:

$$D_4(p, q) = |x-s| + |y-t|$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

c. Satranç Tahtası Uzaklığı:

$$D_8(p, q) = \max(|x-s|, |y-t|)$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

## Soru 5

- Aşağıda verilen piksel düzenlemesindeki daire içerisindeki iki nokta arasındaki satranç tahtası uzaklığı değeri nedir?

0	0	0	0	0
0	0	1	1	0
0	1	1	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0

## Soru 6

- Aşağıda verilen piksel düzenlemesindeki daire içerisindeki iki nokta arasındaki şehir-blok uzaklığı değeri nedir?

0	0	0	0	0
0	0	1	1	0
0	1	1	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0

# Sayısal Görüntü İşlemede Kullanılan Matematiksel İşlemlere Giriş

## ► Dizi ve Matris İşlemleri

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Dizi  
çarpım  
operatörü

$$A .* B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

**Dizi çarpımı**

Matris  
çarpım  
operatörü

$$A * B = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

**Matris çarpımı**

# Sayısal Görüntü İşlemede Kullanılan Matematiksel İşlemlere Giriş

## ► Doğrusal ve Doğrusal Olmayan İşlemler

$$H[f(x, y)] = g(x, y)$$

► Eğer

$$H[a_i f_i(x, y) + a_j f_j(x, y)]$$

$$= H[a_i f_i(x, y)] + H[a_j f_j(x, y)]$$

$$= a_i H[f_i(x, y)] + a_j H[f_j(x, y)]$$

$$= a_i g_i(x, y) + a_j g_j(x, y)$$

**Toplanabilirlik**

**Homojenlik**

ise H'a **doğrusal operatör** denir. Şayet H yukarıdaki niteliği sağlamıyorsa **doğrusal olmayan operatör** olarak ifade edilir.



# Aritmetik İşlemler

- Görüntüler arasındaki aritmetik dizi işlemleridir. Dört aritmetik işlem şu şekildedir:

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

$$d(x,y) = f(x,y) - g(x,y)$$

$$p(x,y) = f(x,y) \times g(x,y)$$

$$v(x,y) = f(x,y) \div g(x,y)$$

## Örnek: Gürültü Azaltma İçin Gürültülü Görüntülerin Toplanması

Gürültüsüz görüntü:  $f(x,y)$

Gürültü:  $n(x,y)$  (her  $(x,y)$  koordinat ikilisinde gürültünün ilintisiz ve sıfır ortalama değere sahip olduğu varsayılmaktadır)

Bozulmuş görüntü:  $g(x,y)$

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

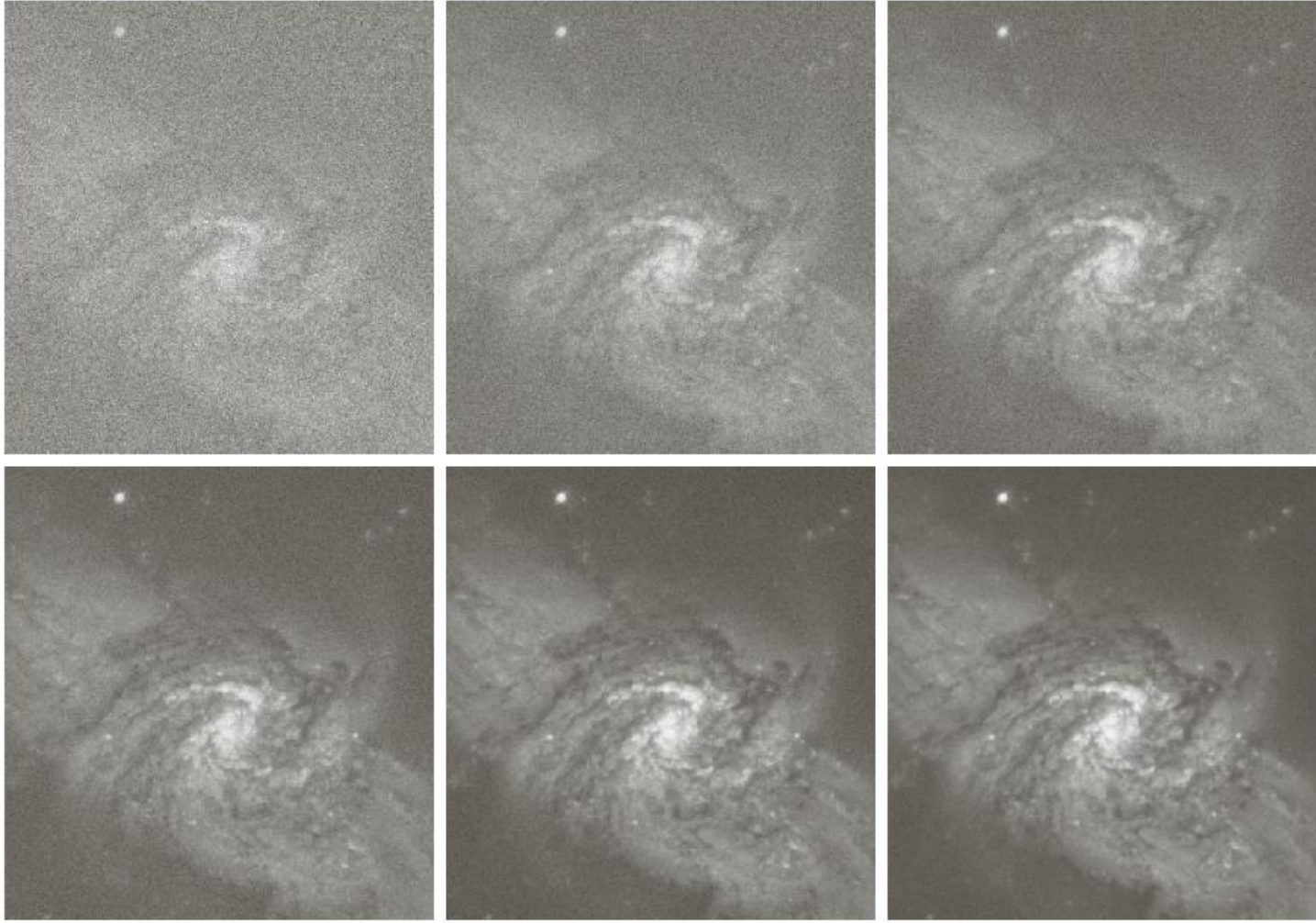
Gürültülü bir görüntü kümesini,  $\{g_i(x,y)\}$  toplayarak gürültü içeriğini azaltmak:

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y)$$

# Örnek: Gürültü Azaltma İçin Gürültülü Görüntülerin Toplanması

- ▶ Astronomide, çok düşük ışık seviyeleri altında yapılan görüntüleme çoğu kez algılayıcı gürültüsünün oluşmasına yol açar. Bu da görüntülerin analiz için işlenmesini neredeyse işe yaramaz hale getirir.
- ▶ Astronomik gözlemlerde, benzer algılayıcıların gürültüyü azaltma amacıyla yeteneklerini birleştirmeyi kullanmasıdır. Bu algılayıcılarda gürültü azaltma, uzun bir süre boyunca aynı yeri gözlemleyerek yapılmaktadır.

# Örnek: Gürültü Azaltma İçin Gürültülü Görüntülerin Toplanması



a b c  
d e f

**ŞEKİL 2.26** (a) Toplanır Gauss gürültüsüyle bozulmuş Galaxy Pair NGC3314 görüntüsü. (b)-(f) Sırasıyla 5,10,20,50 ve 100 tane gürültülü görüntünün ortalamasının alınmasıyla elde edilen sonuçlar.

## Bir Görüntü Çıkarma Örneği: Maske Modlu Radyografi

**Mask  $h(x,y)$ :** hasta vücudunun bir bölgesinin bir X-ışını görüntüsü

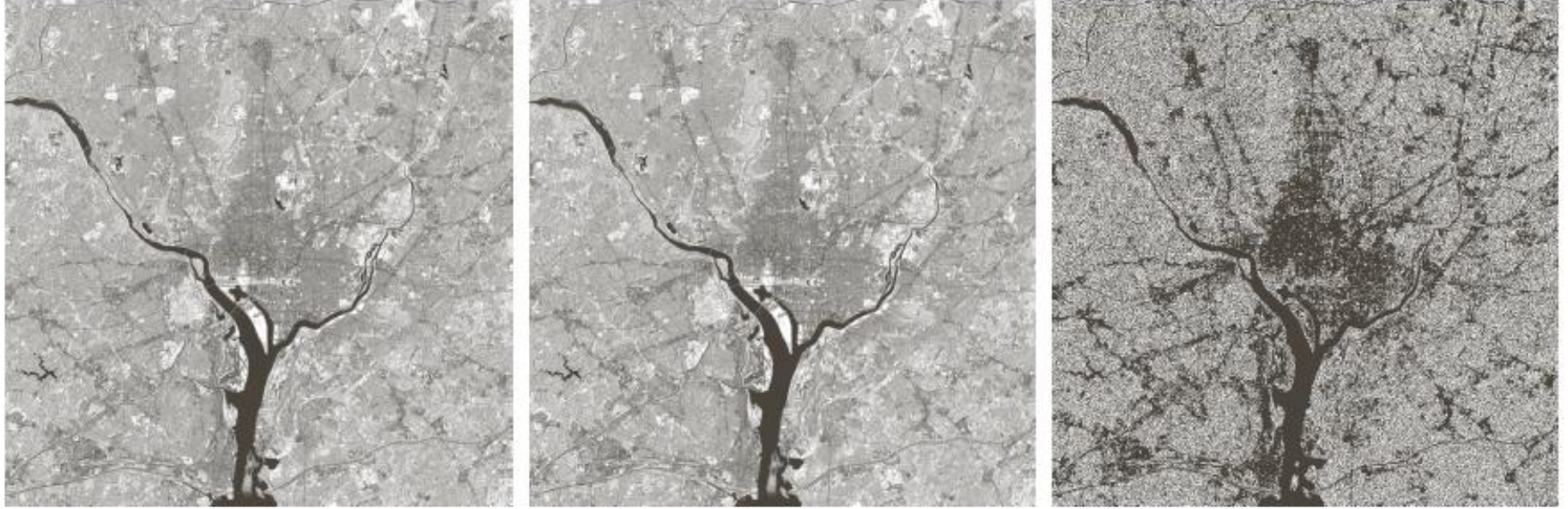
**Canlı görüntüler  $f(x,y)$ :** TV kamerası ile yakalanan bir X-ışını görüntüsü

**Güçlendirilmiş detaylar  $g(x,y)$**

$$g(x,y) = f(x,y) - h(x,y)$$

Görüntüler TV hızında alındığı için bu yöntem esas itibariyle kontrast maddenin gözlemlenen alandaki çeşitli atardamarlarda nasıl yayıldığını gösteren bir film oluşturur.

# Bir Görüntü Çıkarma Örneği: Maske Modlu Radyografi

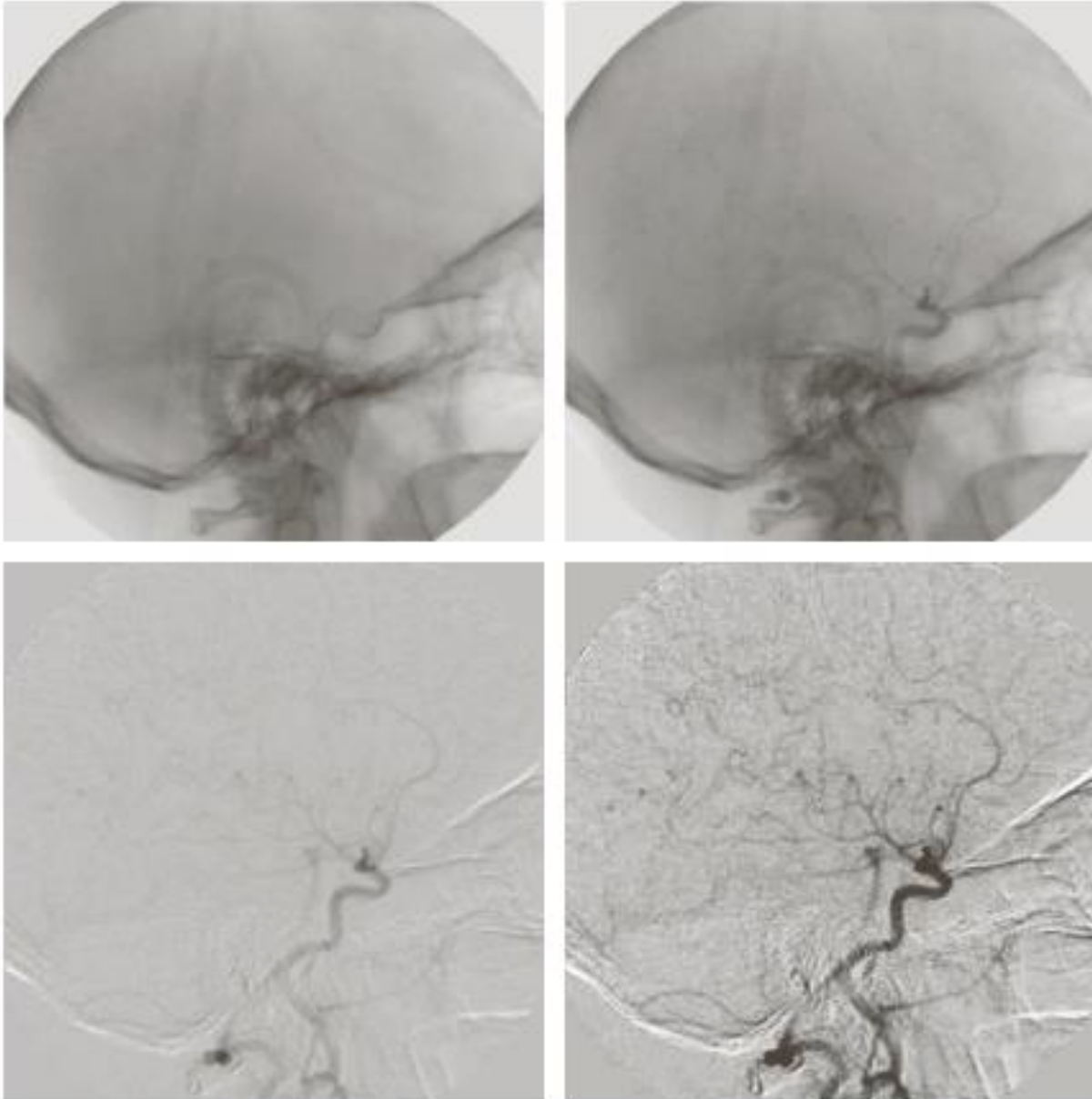


a b c

**ŞEKİL 2.27** (a) Washington D.C. bölgesinin kızıl ötesi görüntüsü. (b) (a)'daki herbir pikselin er önemsiz bitini sıfır yaparak elde edilen görüntü. (c) İki görüntünün farkı. Netlik için  $[0,255]$  aralığında ölçekleme yapılmıştır.



# Bir Görüntü Çıkarma Örneği: Maske Modlu Radyografi

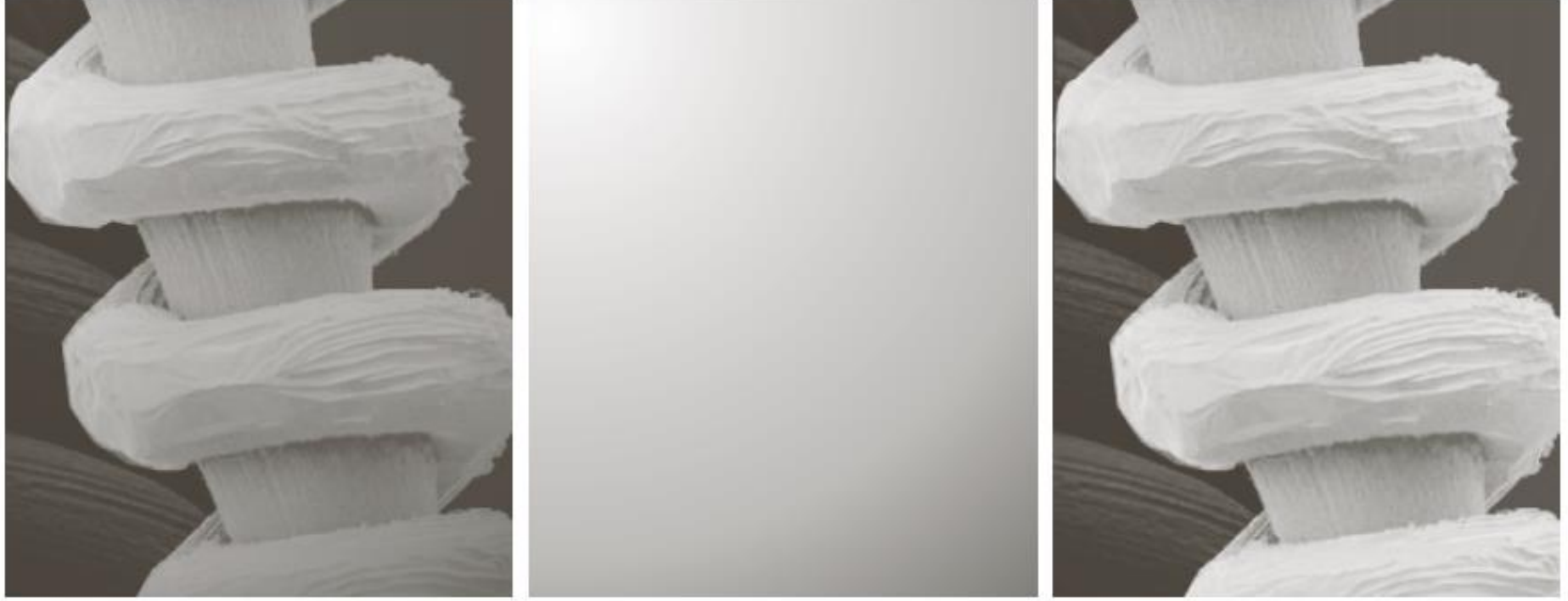


a	b
c	d

## ŞEKİL 2.28

Sayısal çıkarma anjiyografi (a)  
Maske görüntü (b) Canlı görüntü (c) (a) ve (b) arasındaki fark (d) Zenginleştirilmiş fark görüntüsü (Şekil (a) ve (b), The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands izniyle)

# Bir Görüntü Çarpma Örneği



a b c

**ŞEKİL 2.29** Gölge düzeltme (a) Tungsten telinin gölgelenmiş SEM görüntüsü. Yaklaşık 130 kat büyütülmüş hali. (b) Gölge görüntüsü (c) (a)'nın (b)'nin tersi ile çarpımı (Orjinal görüntü Michael Shaffer'in izniyle, Department of Geological Sciences, University of Oregon, Eugene)



# Bir Görüntü Çarpma Örneği



a b c

**ŞEKİL 2.30** (a) X ışını sayısal diş görüntüsü (b) Dolgulu dişleri ayırt etmek için kullanılan ROI maskesi (beyaz bölgeler 1'e siyah bölgeler 0'a karşılık gelmektedir) (c) (a) ve (b)'nin çarpımı.

# Python Görüntü İşleme

► Görüntü İşleme Araç Kutusu, Python'ın sayısal bilgi işlem ortamının yeteneklerini genişleten işlevler topluluğudur. Araç kutusu, aşağıdakiler dahil olmak üzere çok çeşitli görüntü işleme operasyonlarını destekler:

- Geometrik işlemler
- Mahalle ve blok işlemleri
- Doğrusal filtreleme ve filtre tasarımı
- Dönüşümler
- Görüntü analizi ve geliştirme
- İkili görüntü işlemleri
- İlgili bölgesi operasyonları

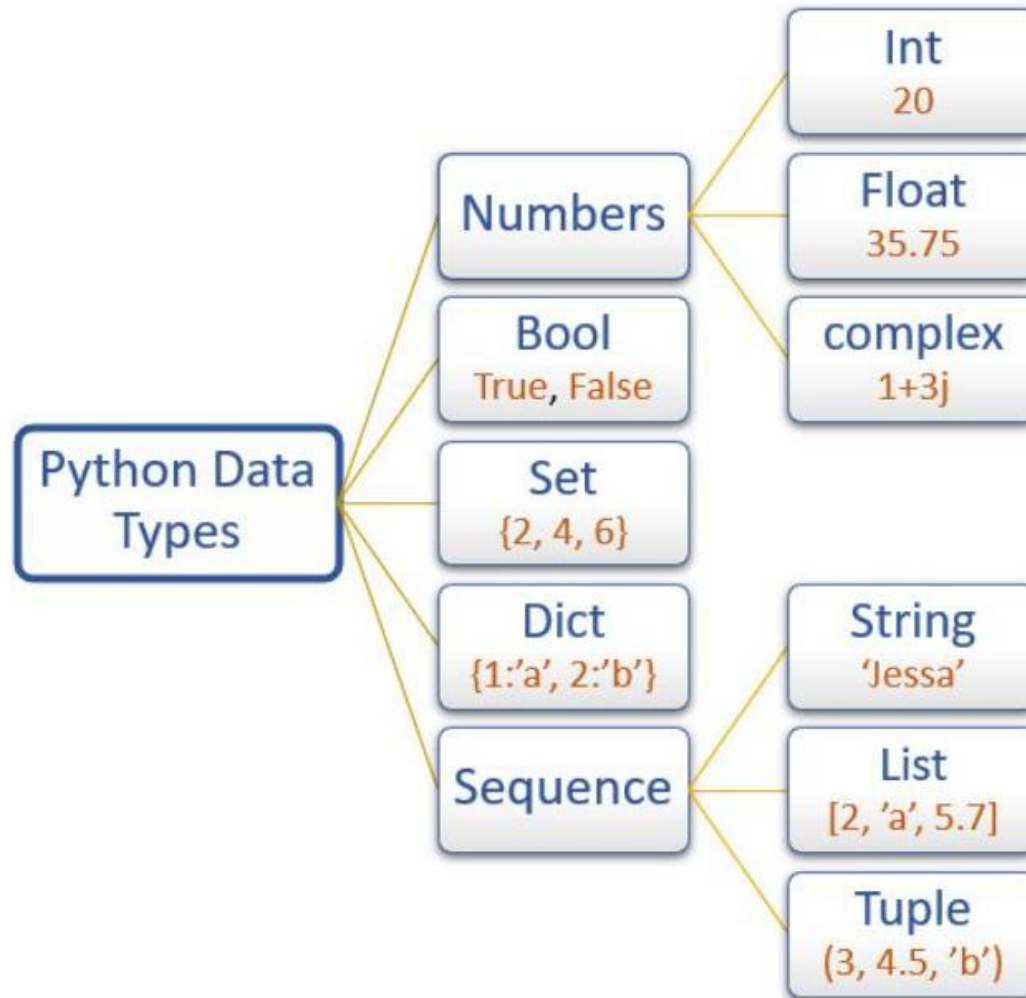
# Python Görüntü İşleme

► Python birçok görüntü formatını içe / dışa aktarabilir:

- BMP (Microsoft Windows Bitmap)
- GIF (Graphics Interchange Files)
- HDF (Hierarchical Data Format)
- JPEG (Joint Photographic Experts Group)
- PCX (Paintbrush)
- PNG (Portable Network Graphics)
- TIFF (Tagged Image File Format)
- XWD (X Window Dump)
- raw-data ve diğer görüntü verisi tipleri

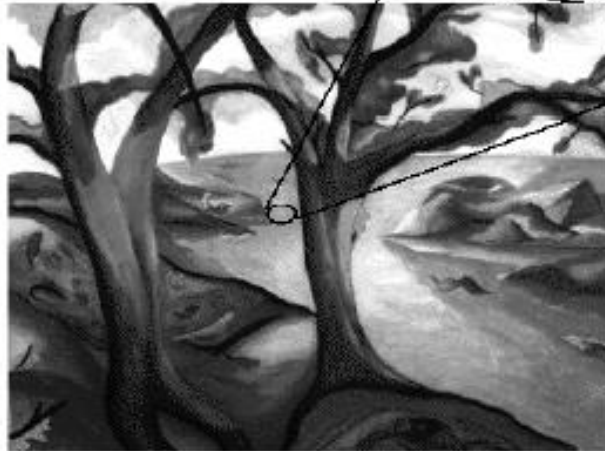
# Python Görüntü İşleme

## ► Python veri tipleri:



# Python Görüntü İşleme

- Binary images :  $\{0,1\}$
- Intensity images :  $[0,1]$  or `uint8`, `double` etc.
- RGB images :  $m \times n \times 3$
- Multidimensional images:  $m \times n \times p$  ( $p$  is the number of layers)



0.2251	0.2553
0.5342	0.2051
0.5342	0.1789
0.4308	0.2483
0.344	0.2624

[illegible]

# Python Görüntü İşleme

## ► Python'da görüntüleri okuma ve yazma

```
ide + Markdown | ▶ Run All ☰ Clear Outputs of All C
```

```
import cv2
import numpy as np

resim1 = cv2.imread("brain.jpg")

cv2.imshow("Brain",resim1)

#Resmin boyutunu öğrenmek için:
print(resim1.size)

#Resmin veri tipini öğrenmek için:
print(resim1.dtype)

#Resmin genişliğini, yüksekliğini ve kaç kanaldan oluştuğunu öğrenmek için:
print(resim1.shape)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

🔄 17.9s



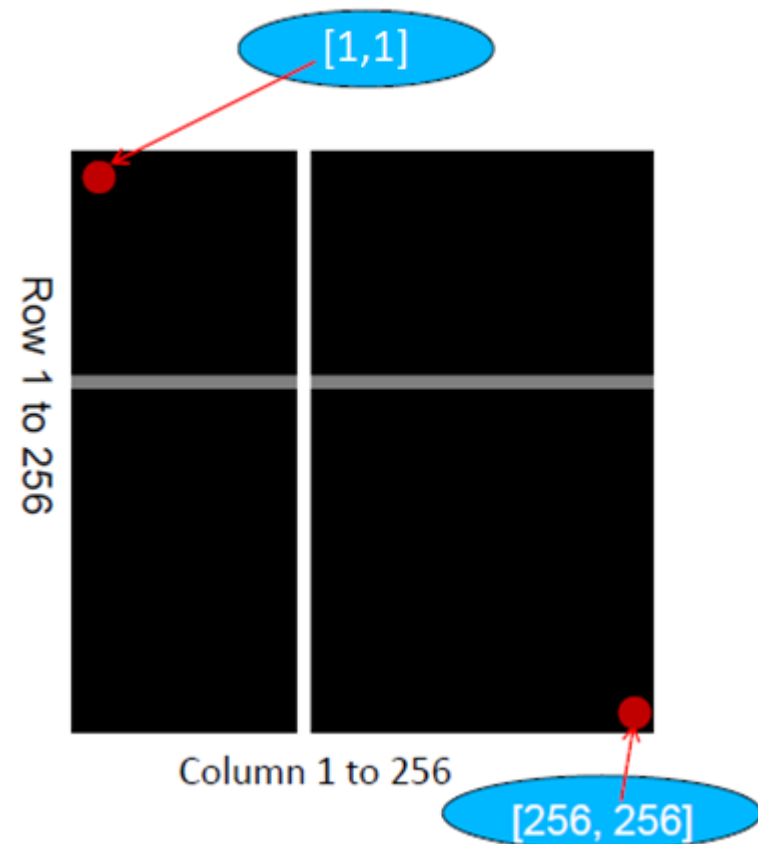
```
230400
uint8
(240, 320, 3)
```

# Python Görüntü İşleme

- Bir matris (veya görüntü) nasıl oluşturulur? Yoğunluk resmi:

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

row = 256
col = 256
img = np.zeros((row,col))
img[100:105, :] = 0.5
img[:, 100:105] = 1
plt.figure(figsize=(10,4))
plt.imshow(img)
plt.show()
```



# Python Görüntü İşleme

## ► İkili görüntü

```
import cv2
import numpy as np

height = 512
width = 512
img = np.random.randint(255, size=(height, width, 1), dtype=np.uint8)

cv2.imshow('Binary',img)
```



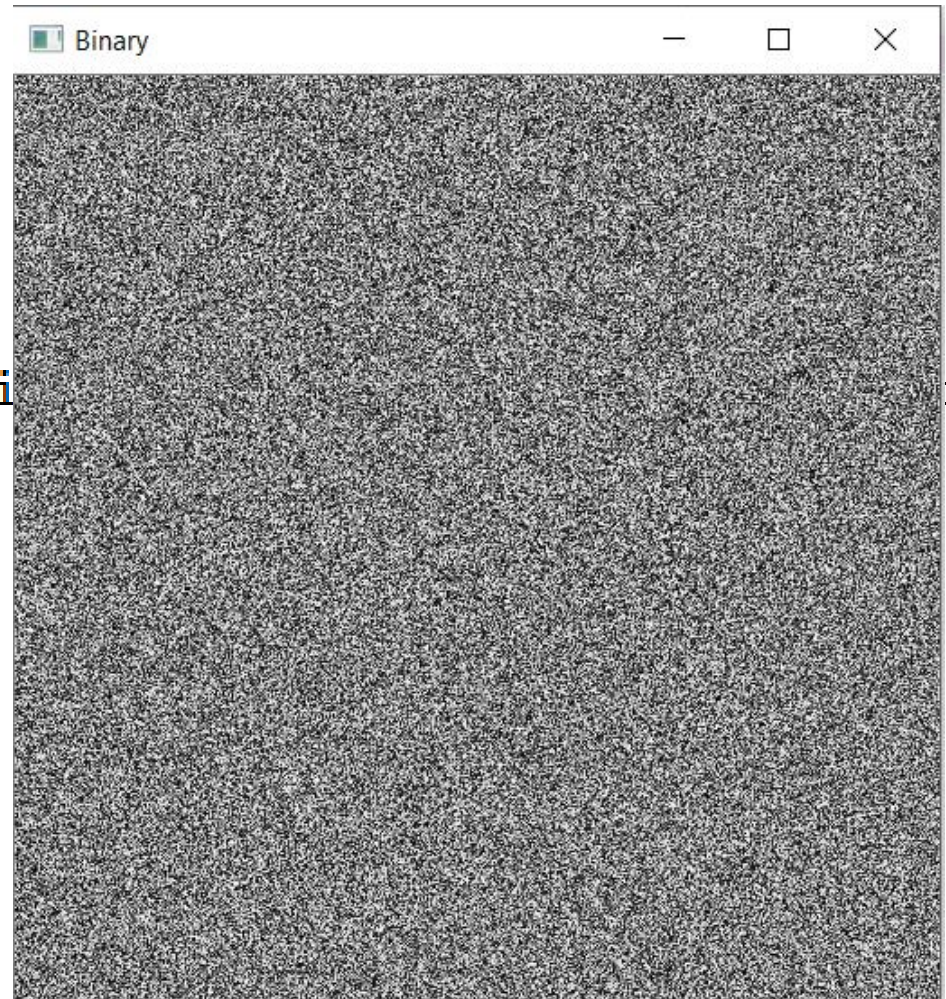
# Python Görüntü İşleme

## ► İkili görüntü

```
import cv2
import numpy as np

height = 512
width = 512
img = np.random.randint(255, size=(height, width, 3))

cv2.imshow('Binary',img)
```



# Python Görüntü İşleme

## ► Vektörlerde indeksleme

>> v = [1, 3, 5, 7, 9] vektörü yazılıp enter'a vurulursa

v =

[1, 3, 5, 7, 9]

>> value=v[2] yazılıp enter'a vurulursa v vektörünün ikinci elemanı value değişkenine atanır:

value =

5

Bir satır vektör bir sütun vektöre transpoze fonksiyonu ile döndürülür:

>> w = np.transpose(v)

w =

1

3

5

7

9

# Python Görüntü İşleme

## ► Vektörlerde indeksleme

>> value = v[1:3] yazılıp enter'a vurulursa v vektörünün 1. indeksten 3. indekse kadar olan elemanları alınır:

value =

**[ 3, 5 ]**

v vektörünün 2'nciden 4'üncü elemana kadar alınmak istenirse:

>> value = v[2:4] yazılmalıdır. Sonuç:

value =

**[ 5, 7 ]**

v vektörünün 3'üncü elemandan sonuna kadar olan elemanların tümü alınmak istenirse:

>> value = v[3:] yazılmalıdır. Sonuç:

value =

**[ 7, 9 ]**

# Python Görüntü İşleme

## ► Vektörlerde indeksleme

Vektörün ilk elemanından başlayıp, birer atlayarak sona kadar tüm elemanları istendiğinde aşağıdaki komutla şu sonuç elde edilir:

```
>> value = v[0::2]
```

```
value =
```

```
[1, 5, 9]
```

>> `v[::-2]` komutu ise vektörün elemanlarını son elemandan ilk elemana birer atlayarak tersten atar. Sonuç:

```
value =
```

```
[9, 5, 1]
```

olacaktır.

# Python Görüntü İşleme

## ► Matrislerde indeksleme

```
A = [[1, 4, 5],    # iç içe liste
      [-5, 8, 9],
      [6, 8, 10],
      [0, 2, 38]]

print("A =", A)
print("A[1] =", A[1])    # 2.satır
print("A[1][2] =", A[1][2]) # 2.satırın 3.elemanı
print("A[0][-1] =", A[0][-1]) # İlk satırın sonuncu elemanı
```

```
A = [[1, 4, 5], [-5, 8, 9], [6, 8, 10], [0, 2, 38]]
A[1] = [-5, 8, 9]
A[1][2] = 9
A[0][-1] = 5
```

# Python Görüntü İşleme

- ▶ Matrislerde indeksleme
  - Numpy kütüphanesi ile de matris oluşturulabilir.

```
import numpy as np

a = np.array([[1, 4, 5],    # numpy array
              [-5, 8, 9],
              [6, 8, 10],
              [0, 2, 38]])

print("type:", type(a))
print("a =", "\n", a, "\n")
print("a[1] =", a[1])      # 2.satır
print("a[1][2] =", a[1][2]) # 2.satırın 3.elemanı
print("a[0][-1] =", a[0][-1]) # İlk satırın sonuncu elemanı
```

```
type: <class 'numpy.ndarray'>
```

```
a =
[[ 1  4  5]
 [-5  8  9]
 [ 6  8 10]
 [ 0  2 38]]
```

```
a[1] = [-5  8  9]
```

```
a[1][2] = 9
```

```
a[0][-1] = 5
```

# Python Görüntü İşleme

## ► Matrislerde indeksleme

```
import numpy as np
C = np.array([[1, 1, 2], [3, 5, 3], [5, 6, 9]])
C[:,2]
```

Output: array([2, 3, 9])

- ( : ) operatörü blok halinde elemanları seçer. Yukarıda ki kodda tüm satırlar ama sadece 2. sütun anlamına gelmektedir. Bize sadece 2. sütun elemanlarını verecektir.

- >> C[ : , 2 ] = 1

C matrisinin tüm satırlarındaki sadece ikinci sütununun elemanlarını 1 yapar.

# Python Görüntü İşleme

## ► Matrislerde indeksleme

- Örnek soru: Üçüncü sütunu 0 olan, ama diğer sütunları A'ya eşit olan bir B matrisini nasıl yapabiliriz ??

Cevap: `>> B = A ;`

`>> B[:, 2] = 0 ;`

- Örnek ; `>> sum ( C[ : ] )` ve `sum(C)` ifadeleri aynı sonucu vermektedir. Her ikisi de, her bir sütunun kendi içerisindeki elemanların toplamını vermektedir.

```
C = np.array([[1, 1, 2], [3, 5, 3], [5, 6, 9]])
```

```
s = sum ( C[ : ] )
```

```
print(s)
```

```
a = sum(C)
```

```
print(a)
```

Output:

```
[ 9 12 14] [ 9 12 14]
```



# Python Görüntü İşleme

## ► Görüntü Operasyonları

- RGB resmi gri görüntüye
- Resim yeniden boyutlandır
- Görüntü kırpma
- Görüntü döndürmek
- Görüntü histogramı
- Görüntü histogramı eşitlemesi
- Konvolüsyon

# Kaynaklar

- ▶ Sayısal Görüntü İşleme, Palme Yayıncılık, Üçüncü Baskıdan Çeviri (*Orj: R.C. Gonzalez and R.E. Woods: "Digital Image Processing", Prentice Hall, 3rd edition, 2008*).
- ▶ “Digital Image Processing Using Matlab”, Gonzalez & Richard E. Woods, Steven L. Eddins, Gatesmark Publishing, 2009
- ▶ Ders Notları, MATLAB for Image Processing, CS638-1 TA: Tuo Wang
- ▶ Ders Notları, BIL717-Image Processing, E.Erdem
- ▶ Ders Notları, EBM537-Görüntü İşleme, F.Karabiber
- ▶ <https://docs.opencv.org/>
- ▶ Bekir Aksoy, Python ile İmgeden Veriye Görüntü İşleme ve Uygulamaları, Nobel Akademik Yayıncılık