



Bilgisayar Mühendisliği Bölümü

Bil 1006-Bilgisayar Grafiğine Giriş

Dr. Öğr. Üyesi Serkan Savaş

Morfolojik dönüşümler

- Erozyon (Aşındırma) (Erosion)
- Genişletme (Dilation)
- Açma (Opening)
- Kapama (Closing)
- Morfolojik Gradyan (Morphological Gradient)
- Top Hat
- Black Hat
- Kernel

Morfolojik dönüşümler

Matematiksel morfoloji, lineer olmayan komşuluk işlemlerinde güçlü bir görüntü işleme analizidir. Morfolojik dönüşümler, görüntü şekline dayalı bazı basit işlemlerdir.

Normalde ikili görüntüler üzerinde gerçekleştirilir.

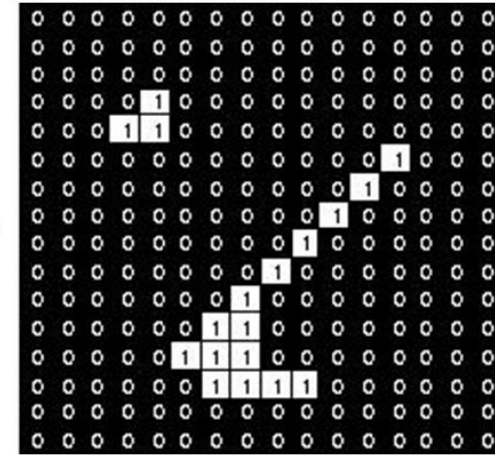
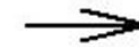
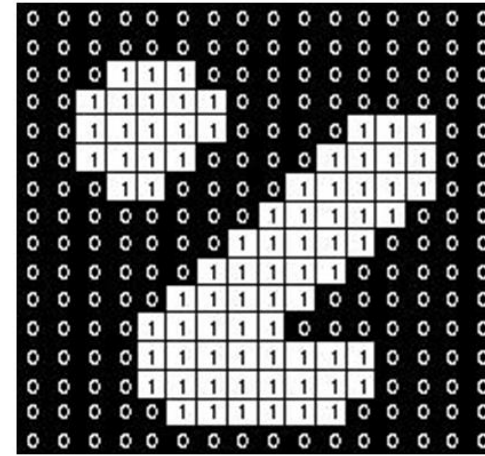
İki girdiye ihtiyaç duyar; biri orijinal görüntü, ikincisi ise işlemin doğasına karar veren yapılandırma elemanı veya çekirdek olarak adlandırılır. İki temel morfolojik operatör **Erozyon** ve **Genişletme**dir. Ardından Açılış, Kapanış, Gradyan vb. varyant formları da devreye girer.

Aşındırma

İkili imgedeki nesneyi küçültmeye ya da inceltmeye yarayan morfolojik işlemdir.

Aşındırma işlemi ile sayısal resim aşındırılmış olur.

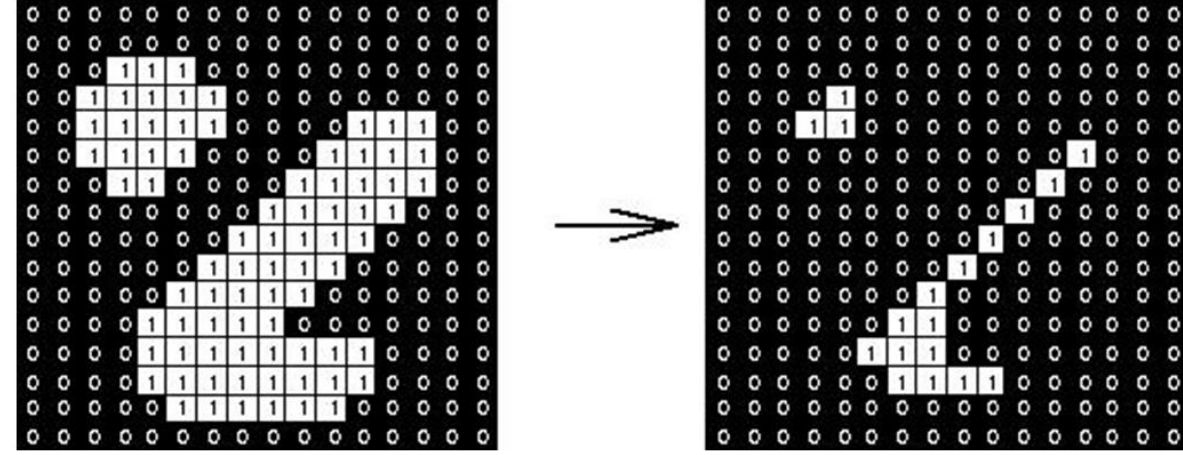
Yani resim içerisindeki nesneler ufalır, delik varsa genişler, bağlı nesneler ayrılma eğilimi gösterir.



Aşındırma

Aşındırma, toprak erozyonu gibidir, ön plan nesnesinin sınırlarını aşındırır.

Çekirdek 2B evrişimdeki gibi görüntü boyunca kayar. Orijinal görüntüdeki bir piksel (1 veya 0), yalnızca çekirdeğin altındaki tüm pikseller 1 ise 1 olarak kabul edilir, aksi takdirde aşınır (sıfır yapılır).



Aşındırma

Sonuçta sınırın yakınındaki tüm pikseller, çekirdeğin boyutuna bağlı olarak atılır.

Böylece görüntüdeki ön plan nesnesinin kalınlığı veya boyutu azalır. Başka bir deyişle sadece beyaz bölge azalır.

Küçük beyaz gürültüleri gidermek, birbirine bağlı iki nesneyi ayırmak vb. için kullanışlıdır.

Aşındırma

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

img = cv2.imread('img/morp.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)

plt.subplot(121), plt.imshow(img, cmap='gray'), plt.title('Orjinal')
plt.xticks([],plt.yticks([]))
plt.subplot(122), plt.imshow(erosion, cmap='gray'), plt.title('Erosion')
plt.xticks([],plt.yticks([]))
```

KKÜ
BiL1006
BGG

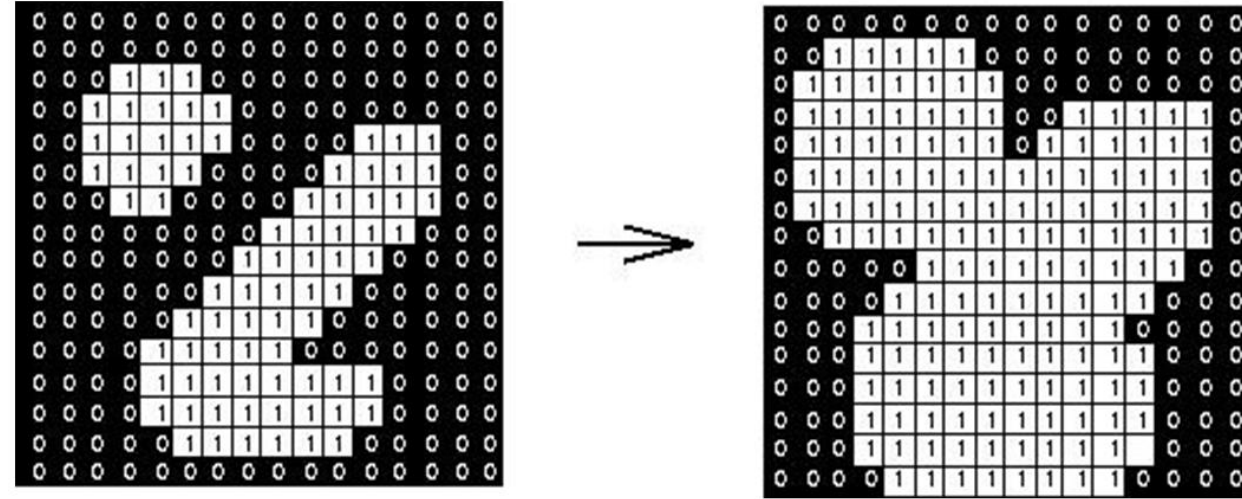
KKÜ
BiL1006
BGG

Geniřletme

İkili imgedeki nesneyi büyütmeye ya da kalınlařtırmaya yarayan morfolojik işlemdir.

Sayısal bir resmi genişletmek resmi yapısal elemanla keřiřtięi bölümler kadar büyütmek demektir.

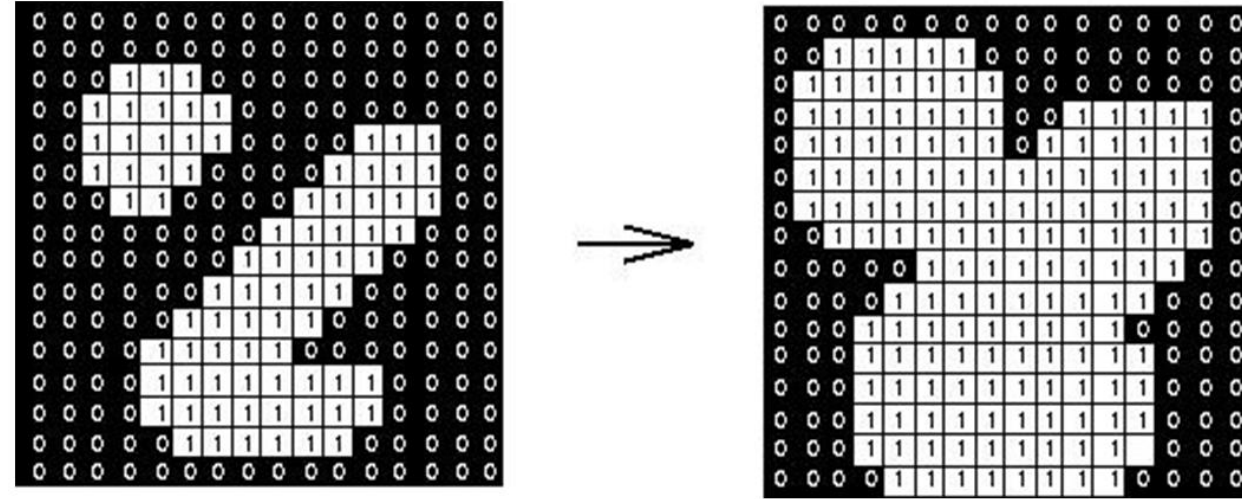
Kalınlařtırma işleminin nasıl yapılacaęını yapı elemanı belirler.



Geniřletme

Ařındırmanın tersidir. Burada, çekirdeğın altındaki en az bir piksel "1" ise bir piksel öğesi "1"dir.

Böylece görüntüdeki beyaz bölge artar veya ön plandaki nesnenin boyutu artar. Normalde gürültü giderme gibi durumlarda aşındırmayı genişleme takip eder.



Geniřletme

Çünkü erozyon beyaz sesleri ortadan kaldırır ama aynı zamanda nesneyi de küçültür. Bu nedenle tekrar genişletme gerekir. Gürültü gittiği için geri gelmeyecektir ama nesne alanı artar. Bir nesnenin kırık parçalarının birleştirilmesinde kullanılabilir.

Genişletme

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

img = cv2.imread('img/morp.png',0)
kernel = np.ones((5,5),np.uint8)
dilation = cv2.dilate(img,kernel,iterations = 1)

plt.subplot(121), plt.imshow(img, cmap='gray'), plt.title('Original')
plt.xticks([],plt.yticks([]))
plt.subplot(122), plt.imshow(dilation, cmap='gray'), plt.title('Erosion')
plt.xticks([],plt.yticks([]))
```

KKÜ
BİL1006
BGG

KKÜ
BİL1006
BGG

Matematiksel modelleri

Geniřletme işleminden sonra resim genelde daha parlaktır.

Aşındırma işleminden sonra resim genelde daha koyudur.

$A \oplus B \leftrightarrow$ Geniřletme

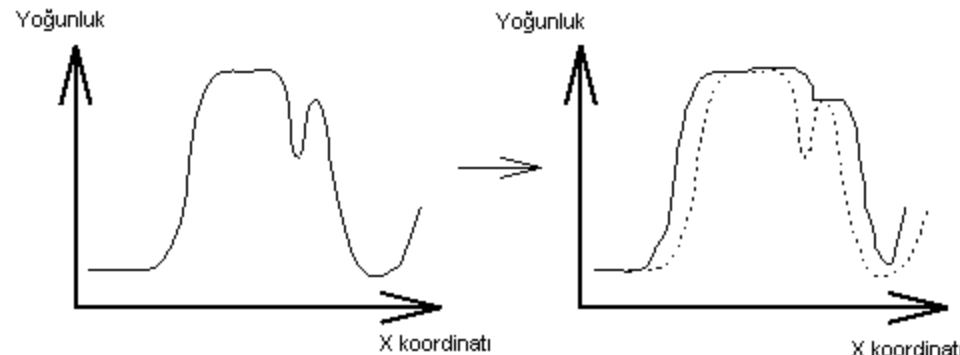
$A \ominus B \leftrightarrow$ Aşındırma

$A \circ B = (A \ominus B) \oplus B \leftrightarrow$ Açma işlemi

$A \bullet B = (A \oplus B) \ominus B \leftrightarrow$ Kapama işlemi

$$(f \oplus b)(x, y) = \max \{f(x - x', y - y') + b(x', y') \mid (x', y') \in D_b\}$$

$$(f \ominus b)(x, y) = \min \{f(x - x', y - y') + b(x', y') \mid (x', y') \in D_b\}$$



Açma ve kapama

$$f \circ b = (f \ominus b) \oplus b \leftrightarrow \text{Açma}$$

$$f \bullet b = (f \oplus b) \ominus b \leftrightarrow \text{Kapama}$$

Açma İşlemi

Bu işlem, genişleme işlemi gerçekleştirdikten sonra yapılan aşındırma işlemidir. Görüntündeki gürültünün giderilmesinde faydalıdır. Bu işlemle birbirine yakın iki nesne görüntüde fazla değişime sebebiyet vermeden ayrılmış olurlar.

Burada **cv2.morphologyEx()** fonksiyonu kullanılır. (cv2.MORPH_OPEN)

Kapama işlemi

Açma işleminin tersidir. Aşındırma ve genişletme işleminin ardışıl uygulanmasıyla da kapama işlemi elde edilir. Dolayısıyla birbirine yakın iki nesne görüntüde fazla değişiklik yapılmadan birbirine bağlanmış olur. Ön plan nesnelerinin içindeki küçük deliklerin veya nesne üzerindeki küçük siyah noktaların kapatılmasında kullanışlıdır.

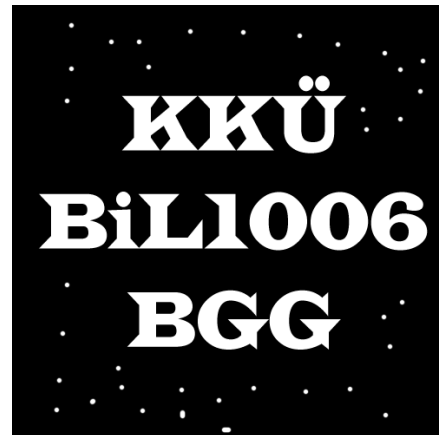
Burada **cv2.morphologyEx()** fonksiyonu kullanılır. (cv2.MORPH_CLOSE)

Açma

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

img2 = cv2.imread('img/op.png',0)
op_kernel = np.ones((7,7),np.uint8)
opening = cv2.morphologyEx(img2, cv2.MORPH_OPEN, op_kernel)

plt.subplot(121), plt.imshow(img2, cmap='gray'), plt.title('Orjinal')
plt.xticks([],plt.yticks([]))
plt.subplot(122), plt.imshow(opening, cmap='gray'), plt.title('Opening')
plt.xticks([],plt.yticks([]))
```



Kapama

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

img3 = cv2.imread('img/cl.png',0)
cl_kernel = np.ones((5,5),np.uint8)
closing = cv2.morphologyEx(img3, cv2.MORPH_CLOSE, cl_kernel)

plt.subplot(121), plt.imshow(img3, cmap='gray'), plt.title('Orjinal')
plt.xticks([],plt.yticks([]))
plt.subplot(122), plt.imshow(closing, cmap='gray'), plt.title('Closing')
plt.xticks([],plt.yticks([]))
```

KKÜ
BİL1006
BGG

KKÜ
BİL1006
BGG

Morfolojik Gradyan

Bir görüntünün genişlemesi ve aşınması arasındaki farktır.

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

img_grd = cv2.imread('img/morp.png',0)
grd_kernel = np.ones((5,5),np.uint8)

gradyan=cv2.morphologyEx(img_grd, cv2.MORPH_GRADIENT, grd_kernel)

plt.subplot(121), plt.imshow(img_grd, cmap='gray'), plt.title('Orjinal')
plt.xticks([], plt.yticks([]))
plt.subplot(122), plt.imshow(gradyan, cmap='gray'), plt.title('Gradyan')
plt.xticks([], plt.yticks([]))
```

Orjinal



Gradyan



Top Hat

Bu işlem giriş görüntüsü ile görüntünün açılması arasındaki farktır. Arka plandan farklı aydınlık seviyeli nesneleri araştıran gri seviyeli resimlerin segmentasyonunda kullanılan bir dönüşümdür.

Gri seviyeli morfolojik işlemler kullanılarak elde edilir. Tepe veya çukur bölgeleri belirginleştirme özelliğine sahiptir.

Aydınlık bölgeler için, $TopHat[A, B] = A - (A \circ B) = A - \max(\min(A))$

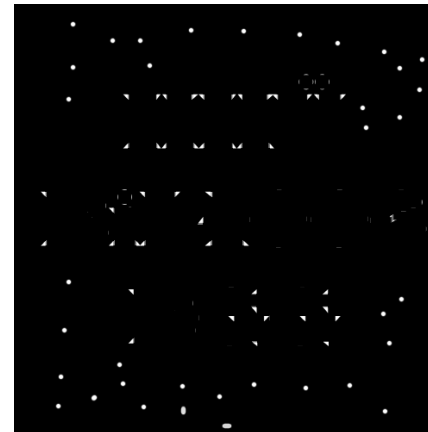
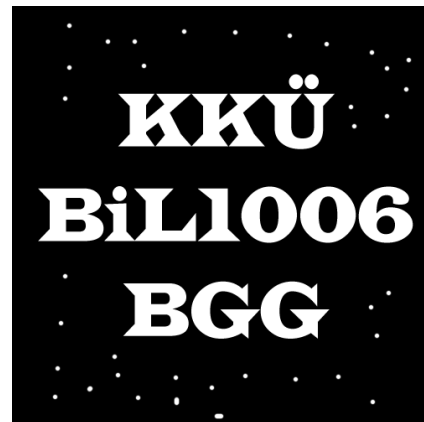
Karanlık bölgeler için, $TopHat[A, B] = (A \bullet B) - A = \min(\max(A)) - A$

Top hat

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

img4 = cv2.imread('img/op.png',0)
top_kernel = np.ones((7,7),np.uint8)
tophat = cv2.morphologyEx(img4, cv2.MORPH_TOPHAT, top_kernel)

plt.subplot(121), plt.imshow(img4, cmap='gray'), plt.title('Orjinal')
plt.xticks([],plt.yticks([]))
plt.subplot(122), plt.imshow(tophat, cmap='gray'), plt.title('TopHat')
plt.xticks([],plt.yticks([]))
```



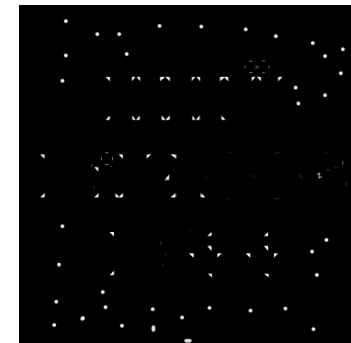
Black Hat

Giriş ile giriş görüntüsünün kapanması arasındaki farktır.

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

img5 = cv2.imread('img/cL.png',0)
black_kernel = np.ones((7,7),np.uint8)
black_hat = cv2.morphologyEx(img5, cv2.MORPH_BLACKHAT, black_kernel)

plt.subplot(121), plt.imshow(img5, cmap='gray'), plt.title('Original')
plt.xticks([],plt.yticks([]))
plt.subplot(122), plt.imshow(black_hat, cmap='gray'), plt.title('BlackHat')
plt.xticks([],plt.yticks([]))
```



Çekirdek

Yapısal eleman olarak da adlandırılan kernel (çekirdek) istenilen boyutlarda ve istenilen şekilde hazırlanmış küçük ikilik bir resimdir. Yapısal eleman çeşitli geometrik şekillerden biri olabilir; en sık kullanılanları kare, dikdörtgen ve dairedir.

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

0	1	0
0	1	1
0	0	0

Çekirdek

Belirlenen kernel yapısı, birçok morfoloji işleminin gerçekleştirilmesinde en önemli öğedir.

Eğer morfolojik işlemin sonucunda resimdeki nesnelerin keskin hatları silinip yerlerine kavisli veya daha yumuşak hatlar getirilmek isteniyorsa dairesel yapısal eleman kullanılabilir.

Örneğin erozyon işleminde resim içerisindeki nesnelerin en ve boyları aynı oranda azaltılmak isteniyorsa yapısal eleman kare seçilmelidir.

Çekirdek

Numpy yardımıyla manuel olarak bir yapılandırma elemanı oluşturulabilir. Örneklerde dikdörtgen şeklinde oluşturduk. Ancak bazı durumlarda eliptik/dairesel şekilli çekirdeklere ihtiyaç olabilir.

Bu amaçla OpenCV'nin **cv2.getStructuringElement()** adlı bir işlevi vardır. Sadece çekirdeğin şekli ve boyutu parametre olarak geçilir ve istenilen çekirdek elde edilir.

Çekirdek

Rectangular Kernel:

`cv2.getStructuringElement(cv2.MORPH_RECT,(5,5))`

Elliptical Kernel:

`cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))`

Cross-shaped Kernel:

`cv2.getStructuringElement(cv2.MORPH_CROSS,(5,5))`

```
rect_kernel= cv2.getStructuringElement(cv2.MORPH_RECT,(5,5))
elips_kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))
cross_kernel =cv2.getStructuringElement(cv2.MORPH_CROSS,(5,5))
```

rect_kernel

```
array([[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]], dtype=uint8)
```

elips_kernel

```
array([[0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0]], dtype=uint8)
```

cross_kernel

```
array([[0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0]], dtype=uint8)
```

Ek kaynaklar

- <https://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm>
- Gonzalez, R. C., & Woods, R. E., (2008), Digital Image Processing, Part 9, 3rd Edition, Pearson International Edition, Pearson Education.

KAYNAKLAR

- Gonzalez, R. C., & Woods, R. E., (2008), Digital Image Processing, 3rd Edition, Pearson International Edition, Pearson Education.
- https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html#morphological-ops
- Boztoprak, H., Çağlar, M. F., & Merdan, M. (2007). Alternatif morfolojik bir yöntemle plaka yerini saptama, XII. *Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Ulusal Kongresi, Eskişehir, Kasım.*