GENEL FONKSİYON ve KOMUTLAR

Resim birleştirme (Yatay)

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
x=cv2.imread('balonlar2.jpg')
y=cv2.imread('anahtar.jpg')
print("x boyutu:"+str(x.shape))
print("y boyutu:"+str(y.shape))
#iki resmi aynı boyutlara getirmeliyiz.
y=cv2.resize(y,(226,223))
#iki resmi yan yana birleştirildi.
z=cv2.hconcat([x,y])
cv2. imshow('yatay birlesik resim',z)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

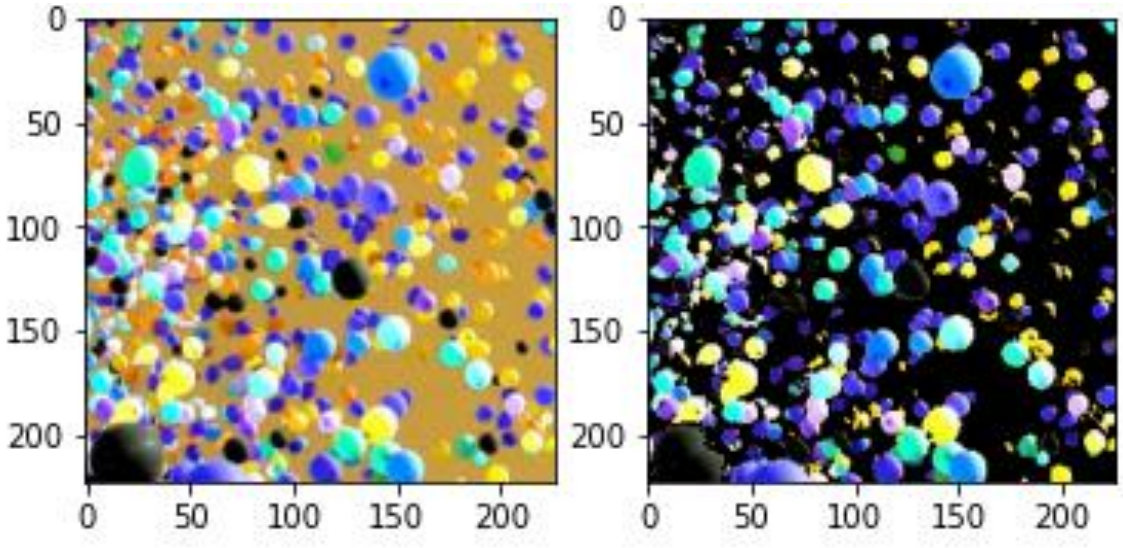
Resim birleştirme (Dikey)

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
x=cv2.imread('balonlar2.jpg')
y=cv2.imread('anahtar.jpg')
print("x boyutu:"+str(x.shape))
print("y boyutu:"+str(y.shape))
#iki resmi aynı boyutlara getirmeliyiz.
y=cv2.resize(y,(226,223))
```

```
#iki resmi üst üste birleştirildi.
z=cv2.vconcat([x,y])
cv2. imshow('yatay birlesik resim',z)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Bir resimde piksel piksel renk aralığı seçme (for döngüsü)

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
sayac=0
x=cv2.imread('balon.jpg')
plt.subplot(121)
plt.imshow(x)
for i in range(x.shape[0]):
   for j in range(x.shape[1]):
      if x[i,j,0]>100 and x[i,j,1]<200 and x[i,j,2]<200:
         x[i,j,:]=0
         sayac=sayac+1
plt.subplot(122)
plt.imshow(x)
```



Yukarıda, belirlenen aralıkta B, G ve R parlaklık değerine sahip olan pikseller siyah yapıldı.

Resim üzerine şekiller çizme

Bir resim üzerine dörtgen, çizgi ve daire çizmek.

```
import cv2
import numpy as np
resim=np.zeros((400,400,3),dtype='uint8')
cv2.rectangle(resim, (10,10), (390,210), (0,255,251),3)
cv2.line(resim,(10,10), (390,210), (0,0,251),3)
cv2.line(resim,(10,230), (390,230), (123,45,78),3)
cv2.circle(resim,(200,350), 25, (148,0,4),3)
cv2. imshow('siyah',resim)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Başlangıç koordinatları

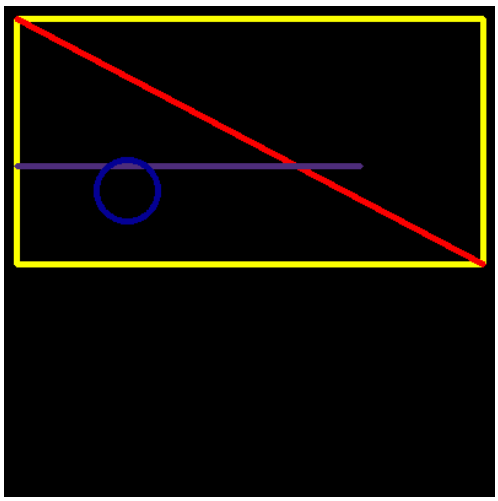
Bitiş koordinatları

Kalınlık

Renk Kodu

Yarıçap

Başlangıç koordinatları



Aritmetik ve Mantıksal işlemler

İki resim ve maske arasında mantıksal işlemler

İki resim seçelim ve bir de siyah-beyaz maske oluşturalım.

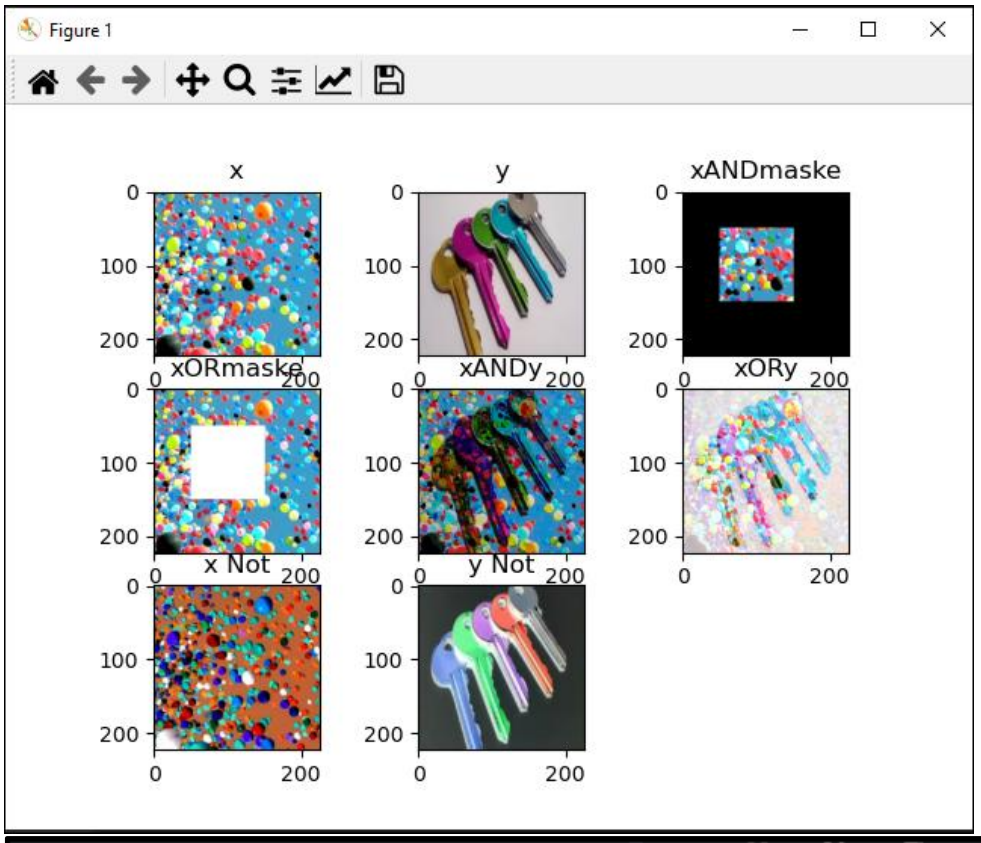
```
import cv2
import numpy as np
from matplotlib import pyplot as plt
x=cv2.imread('balonlar2.jpg')
y=cv2.imread('anahtar.jpg')

print("x boyutu:"+str(x.shape))
print("y boyutu:"+str(y.shape))
#İki resmi aynı boyutlara getirmeliyiz.
y=cv2.resize(y,(226,223))
print("y boyutu:"+str(y.shape))

#Bir maske oluşturalım.
maske=np.zeros((223,226,3), np.uint8)
maske[50:150,50:150]=255

#MANTIKSAL İŞLEMLER
xANDmaske=cv2.bitwise_and(x,maske)
xORmaske=cv2.bitwise_or(x,maske)
xANDy=cv2.bitwise_and(x,y)
xORy=cv2.bitwise_or(x,y)
x_NOT=cv2.bitwise_not(x)
y_NOT=cv2.bitwise_not(y)
maske_NOT=cv2.bitwise_not(maske)

plt.figure(1)
plt.subplot(331)
plt.imshow(cv2.cvtColor(x, cv2.COLOR_BGR2RGB)),plt.title('x')
plt.subplot(332)
plt.imshow(cv2.cvtColor(y, cv2.COLOR_BGR2RGB)),plt.title('y')
plt.subplot(333)
plt.imshow(cv2.cvtColor(xANDmaske, cv2.COLOR_BGR2RGB)),plt.title('xANDmaske')
plt.subplot(334)
plt.imshow(cv2.cvtColor(xORmaske, cv2.COLOR_BGR2RGB)),plt.title('xORmaske')
plt.subplot(335)
plt.imshow(cv2.cvtColor(xANDy, cv2.COLOR_BGR2RGB)),plt.title('xANDy')
plt.subplot(336)
plt.imshow(cv2.cvtColor(xORy, cv2.COLOR_BGR2RGB)),plt.title('xORy')
plt.subplot(337)
plt.imshow(cv2.cvtColor(x_NOT, cv2.COLOR_BGR2RGB)),plt.title('x Not')
plt.subplot(338)
plt.imshow(cv2.cvtColor(y_NOT, cv2.COLOR_BGR2RGB)),plt.title('y Not')
```

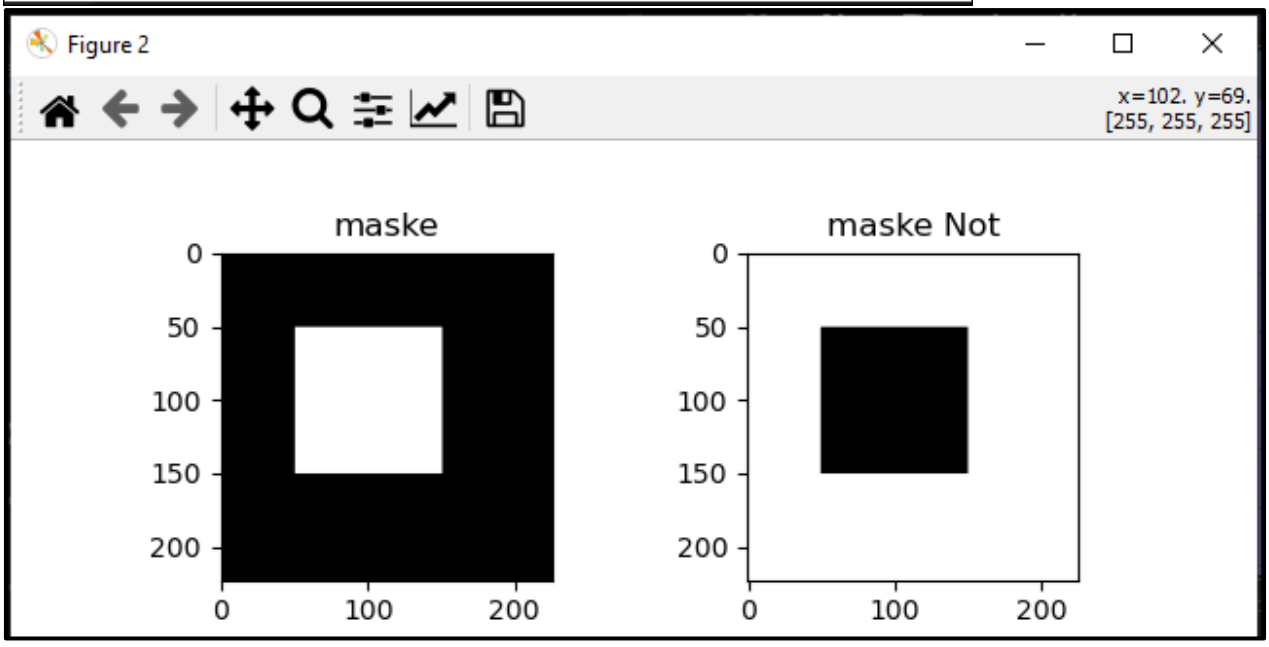
Matplotlib kütüphanesi resimleri RGB formatında görürken **Opencv** kütüphanesi BGR formatında görür.

Resimleri cv2 yani Opencv ile okuduğumuz için **matplotlib** ile görüntüleyeceksek aşağıdaki fonksiyon parametrelerini kullanmalıyız.

plt.imshow(cv2.cvtColor(x, cv2.COLOR_BGR2RGB))

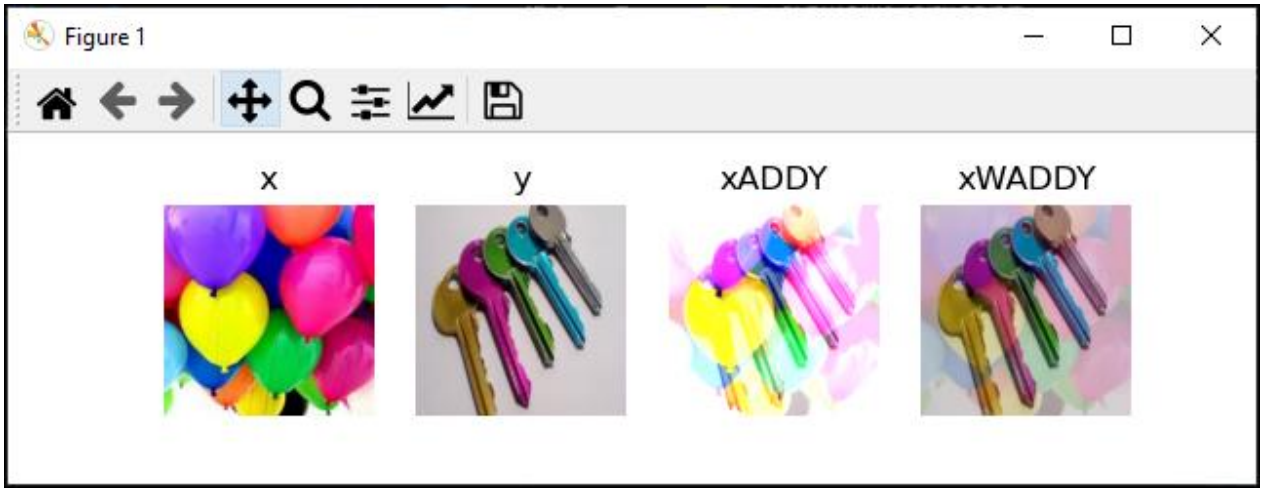
#İkinci figür penceresini açıyoruz.
plt.figure(2)
plt.subplot(221)
plt.imshow(cv2.cvtColor(maske, cv2.COLOR_BGR2RGB)),plt.title('maske')
plt.subplot(222)
plt.imshow(cv2.cvtColor(maske_NOT, cv2.COLOR_BGR2RGB)),plt.title('maske Not')





Resimlerin toplanması ve ağırlıklı toplanması

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
x=cv2.imread('balon.jpg')
y=cv2.imread('anahtar.jpg')

print("x boyutu:"+str(x.shape))
print("y boyutu:"+str(y.shape))
#iki resmi aynı boyutlara getirmeliyiz.
y=cv2.resize(y,(225,225))
print("y boyutu:"+str(y.shape))

#Toplama işlemi
xADDy=cv2.add(x,y)
#Ağırlıklı toplama işlemi
xWADDy=cv2.addWeighted(x,0.2, y, 0.8,0)

plt.figure(1)
plt.subplot(141)
plt.imshow(cv2.cvtColor(x, cv2.COLOR_BGR2RGB)),plt.title('x'),plt.axis("off")
plt.subplot(142)
plt.imshow(cv2.cvtColor(y, cv2.COLOR_BGR2RGB)),plt.title('y'),plt.axis("off")
plt.subplot(143)
plt.imshow(cv2.cvtColor(xADDy, cv2.COLOR_BGR2RGB)),plt.title('xADDY'), plt.axis("off")
plt.subplot(144)
plt.imshow(cv2.cvtColor(xWADDy, cv2.COLOR_BGR2RGB)),plt.title('xWADDY'),plt.axis("off")
```



Resim bulanıklaştırma

```
x = cv2.imread('balon.jpg')
# Resmi bulanıklaştır
bulanikResim = cv2.GaussianBlur(x, (15, 15), 0)
# Sonucu göster
cv2.imshow('Blurred Image', bulanikResim)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

(15, 15): Bu, bulanıklığın genişliğini ve yüksekliğini belirten filtre boyutudur. Burada, her iki boyut da 15'tir, bu da 15x15 boyutunda bir Gauss filtresi kullanılacağı anlamına gelir. Bu değer ne kadar büyük olursa, bulanıklık o kadar güçlü olur.

0: Bu, Gauss çekirdeği içindeki standart sapmanın x ve y yönlerindeki değeridir.

Resim Kenarlarını Çıkartma

```
import cv2
# Resmi yükle
x = cv2.imread('anahtar.jpg')
# Kenarları belirginleştirme
kenar = cv2.Canny(x, 100, 200)
# Sonucu göster
cv2.imshow('Kenarlar, kenar)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

100: İlk eşik değeri. Bu, kenarlar olarak kabul edilen piksellerin yoğunluğunu belirler. Eğer bir pikselin yoğunluğu bu değerin üstünde ise, kenar olarak kabul edilir.

200: İkinci eşik değeri. Bu, kenarlar arasında bağlantı kurulabilmesi için kullanılan bir eşik değeridir. Yüksek bir değer, daha düzgün kenarlar elde etmenizi sağlar, ancak düşük bir değer daha fazla kenar tespiti yapabilir.