GÖRÜNTÜLERDEKİ NESNELERİN TESPİTİ

Görüntü işlemede nesne tespiti, dijital görüntüler üzerinde belirli nesnelerin konumunu ve sınıfını otomatik olarak tespit etme sürecidir. Bu süreç, bilgisayarın bir görüntüde belirli nesneleri tanıması ve yerlerini belirlemesi anlamına gelir.

Nesne tespiti, aşağıda verilen uygulamalarda sıklıkla kullanılmaktadır.

- Üretim hattındaki nesneleri takip etmek, hatalı ürünleri tanımlamak,
- Güvenlik kameraları ve gözetim sistemleri ile tehlikeli durumlar tespit edilebilir. Örneğin, belirli bir bölgeye giren veya çıkan nesneleri izlemek,
- Kavşaklar ve yollar üzerindeki trafiği izlemek ve yönetmek,
- Tıbbi görüntülerde organları, tümörleri veya diğer önemli yapıları tespit etmek,
- Otonom araçlar ve insansız hava araçları (İHA) gibi sistemlerde nesne tespiti, çevrelerindeki nesneleri algılama sayesinde güvenli sürüş veya saldırgan araç tespiti,
- Tarım sektöründe nesne tespiti, bitki hastalıklarını tespit etmek, ürünleri sınıflandırmak ve hasat süreçlerini optimize etmek

Dersimiz kapsamında bir görüntüdeki nesneleri tespit ederken izleyeceğimiz yol:

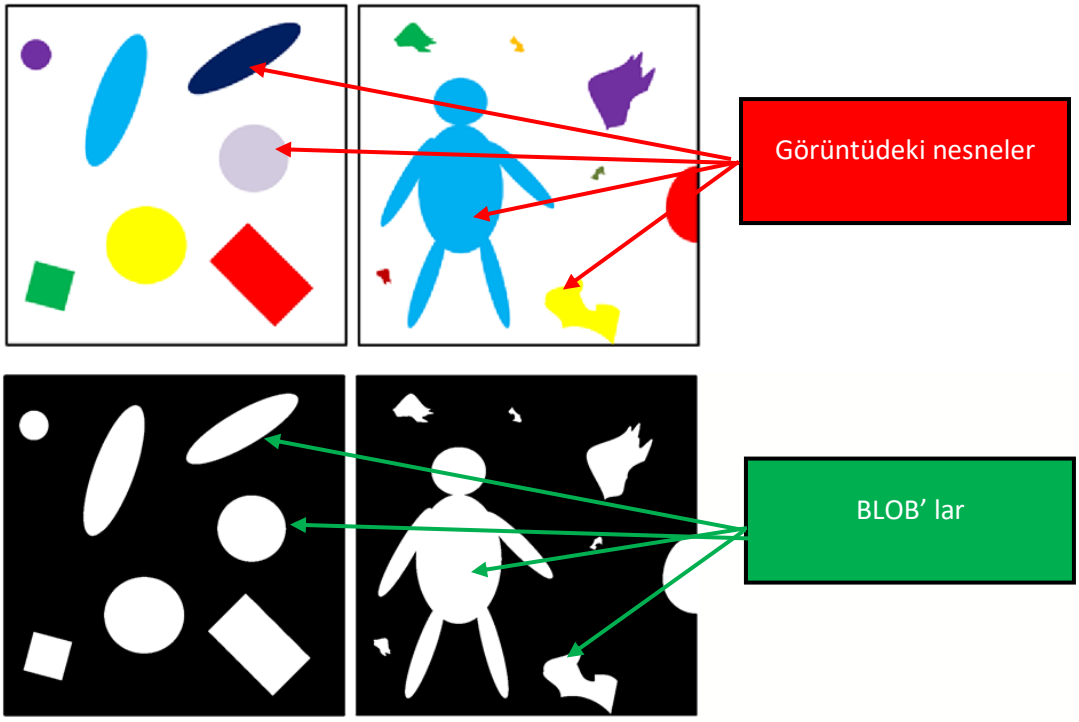
- Önce nesneleri arka plandan ayırmak için çeşitli eşikleme yöntemleri ve eşik değerler kullanarak ikilik (siyah/beyaz) görüntüler oluşturmak, (Örneğin, bir otonom araç için, yol arka plandır ve yol üzerindeki yaya bir nesnedir. İkilik görüntüde yolu siyah ve yayayı da beyaz yapmaya çalışacağız. Beyaz yaptığımız nesne BLOB olarak adlandırılmaktadır.)
- Sonra, bulduğumuz nesneyi etiketleyip işaretleyeceğiz. Hangisi nesne, hangisi gürültü sınıflandıracağız.

BLOB Analizi:

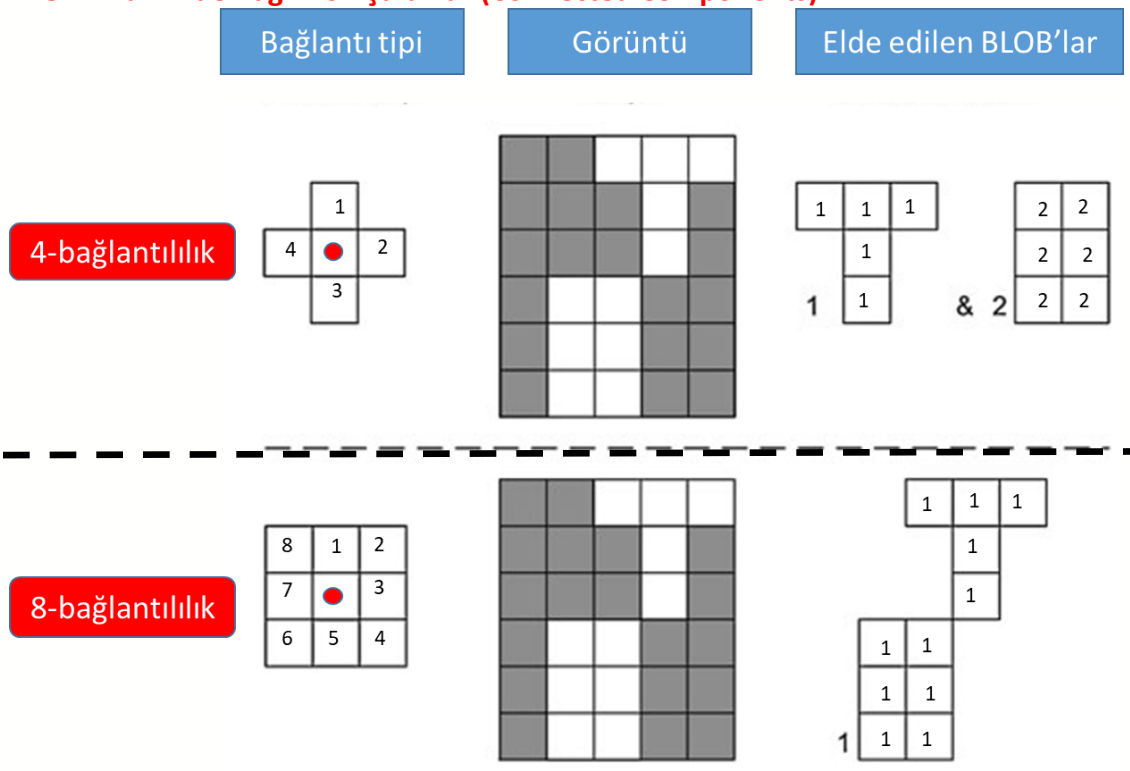
BLOB (Binary Large Object) ikili büyük nesne anlamına gelir ve bir ikili görüntüde bağlı piksellerin bir grubunu ifade etmektedir. Büyük terimi belirli boyuttaki nesne olarak adlandırılır. Dolayısıyla büyük boyutun dışında kalan küçük nesneler gürültü olarak değerlendirilir. Görüntü işlemede nesne tespit etme konusunda özellikle yaygın kullanılır. Bir görüntüde odaklandığımız nesneleri genellikle önce ikilik görüntüye çevirip BLOB haline getiririz.

BLOB analizinin amacı: Bilgisayar görmesi, insan bilgisayar etkileşimi veya örüntü tanıma için nesnelerin etiketlenmesini ya da başka bir ifade ile ikili görüntüdeki büyük nesneleri diğerlerinden (gürültüden) ayırıp etiketleyip öznitelik verileri üretmektir.

Aşağıdaki şekillerde bir görüntüdeki nesnelerin ikilik görüntü (sitah/beyaz) BLOB'lara dönüştürülmesinin bir örneği verilmiştir.



BLOB Analizinde Bağlı Komşuluklar (Connected Components):



BLOB çıkarmanın amacı, ikili bir görüntüdeki BLOB'ları (nesneleri) izole etmektir. Yukarıda belirtildiği gibi, bir BLOB bir grup **bağlı pikselden** oluşur. İki pikselin bağlantılı olup olmadığı, bağlantı, yani hangi piksellerin komşu olduğu ve hangilerinin olmadığı ile tanımlanır. En sık uygulanan iki bağlantı türü Yukarıdaki şekilde gösterilmektedir. 8-bağlantılılık 4-bağlantılılıktan daha doğrudur, ancak 4-bağlantılılık daha az hesaplama gerektirdiği ve dolayısıyla görüntüyü daha hızlı işleyebildiği için sıklıkla uygulanır. İki farklı bağlantı türünün etkisi, ikili görüntülerin bağlantıya bağlı olarak bir ya da iki BLOB içerdiği de şekilde gösterilmektedir.

Aşağıda bir görüntüdeki BLOB'ları tespit eden ve kapsayıcı kutu içine alan program verilmiştir. Ancak daha önce cv2.connectedComponentsWithStats() fonksiyonunu açıklayalım:

cv2.connectedComponentsWithStats() fonksiyonu, bir görüntüdeki bağlı bileşenleri (connected components) etiketleme ve bu bileşenlerle ilgili istatistiksel bilgileri elde etmek için kullanılır. Bu fonksiyon, genellikle nesne tanıma, görüntü analizi ve etiketleme uygulamalarında kullanılır.

Kullanımı:

retval, labels, stats, centroids = cv2.connectedComponentsWithStats(image, connectivity, ltype)

Argümanlar:

image: Etiketleme yapılacak olan giriş görüntüsü (genellikle ikili veya gri tonlamalı).

connectivity: Bileşenleri bağlılıkları açısından tanımlayan bir parametre. Örneğin, 8 bağlılık için 8 veya 4 bağlılık için 4 kullanılabilir.

ltype: Çıktı etiketleme türü. Genellikle cv2.CV_32S kullanılır.

Çıktılar:

retval: Etiket sayısı (arkaplan dahil).

labels: Her pikselin etiketini içeren bir matris.

stats: Bileşen istatistikleri. Her bir bileşen için bir satır ve 5 sütun içerir. Her bir BLOB için, Alan, yükseklik, genişlik, x koordinatı, y koordinatı bilgilerini içerir.

centroids: Bileşen kütle merkezlerini içeren bir dizi.

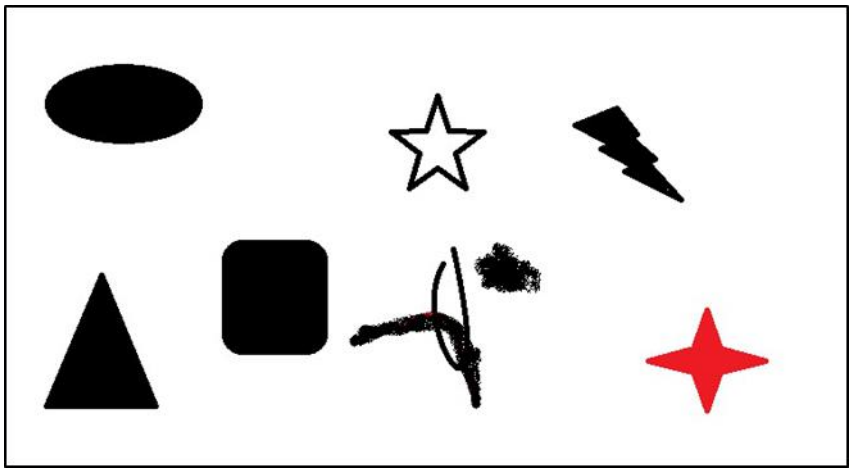
Örnek bir nesne tespit kodu aşağıda verilmiştir.

```
import cv2
from matplotlib import pyplot as plt

x=cv2.imread('deneme2.png')
y=x.copy()
xGri=cv2.cvtColor(x, cv2.COLOR_BGR2GRAY)

xBlur=cv2.blur(xGri, (5,5))
_ ,xEsiklenmis=cv2.threshold(xBlur, 140, 255, cv2.THRESH_BINARY_INV)
# Görüntüdeki nesneler bulunuyor ve ilişkili bilgiler hesaplanıyor.
toplamBlob, etiketler, deger, agrMrk=cv2.connectedComponentsWithStats(xEsiklenmis,4,cv2.CV_32S)

for i in range(1,toplamBlob):
    x1 = deger[i, cv2.CC_STAT_LEFT]
    y1 = deger[i, cv2.CC_STAT_TOP]
    w = deger[i, cv2.CC_STAT_WIDTH]
    h = deger[i, cv2.CC_STAT_HEIGHT]

    #KAPSAYICI KUTU
    solUst = (x1, y1)
    sagAlt = (x1+ w, y1+ h)
   # Herbir blob'a kapsayıcı kutu çiziliyor.
    cv2.rectangle(y,solUst, sagAlt, (255, 0, 0), 2)

plt.subplot(131)
plt.imshow(cv2.cvtColor(y, cv2.COLOR_BGR2RGB)),plt.title('Orjinal'),plt.axis("off")
plt.subplot(132)
plt.imshow(cv2.cvtColor(xBlur, cv2.COLOR_BGR2RGB)),plt.title('Blurlanmış'),plt.axis("off")
plt.subplot(133)
plt.imshow(xEsiklenmis, cmap='gray'),plt.title('İkilik'),plt.axis("off")
```

Döngü içinde,

tüm nesnelerin kapsayıcı kutularının sol üst köşe koordinatları, genişlik ve yükseklikleri alınıyor.

Orijinal Görüntü:



Nesnelerin tespit edildiği görüntü