

GÖRÜNTÜDEKİ NESNELERİN KENARLARINI BULMA

Bu işlemi, görüntü üzerinde nesne tespiti yapmamız gereken durumlarda kullanıyoruz. Fotoğraftaki tüm nesnelerin kenarlarını çıkartıyoruz ve yüz hatlarına uygun kenarlar üzerinde arama yapıyoruz. Böylece işlem yapma süresi kısalmış oluyor.

Görüntüleri siyah beyaz formata çevirdiğimizde sadece parlaklık değeri kalır. Piksel değerlerinde keskin bir geçiş var ise burada kenar var demektir. Eşikleme yöntemini de kullanarak hangilerinin gerçekten kenar, hangilerinin kenar olmadığına karar veririz.

Canny kenar çıkarma sık kullanılan bir algoritmadır.



Maksimum eşik değerinin üzerinde olanlar kesin-kenar olacak; minimum değerler altında kalanlar kenar olarak değerlendirilmeyecek.

UYGULAMA

Metin ve yüz içeren iki görüntüyü inceleme.

Kütüphaneleri ekleme:

```
[1] import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
```

Metin içeren görüntüyü okuma: 0 değerini verirsek görüntüyü siyah beyaz olarak okur. Kenar bulma işleminde renklerle işimiz olmadığı için siyah beyaz görüntü kullanıyoruz.

```
img = cv.imread('sudoku.jpg',0)
kenar = cv.Canny(img,50,90)
```

Görüntüyü ekrana basma:

```
plt.figure(figsize=(12,14))

plt.subplot(121), plt.imshow(img, cmap='gray'), plt.title('Orijinal')
plt.xticks([], plt.yticks([]))

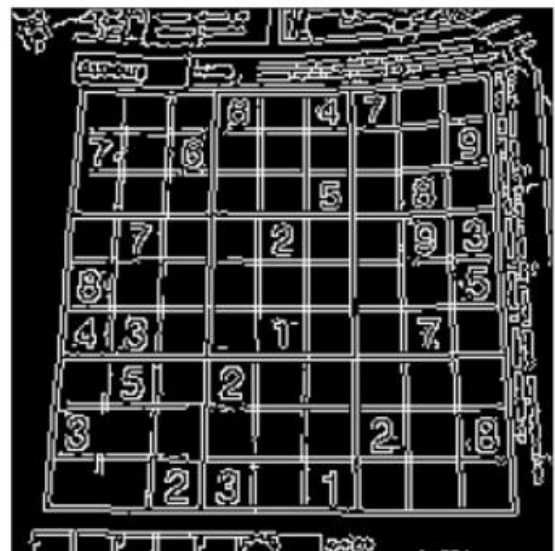
plt.subplot(122), plt.imshow(kenar, cmap='gray'), plt.title('Kenarlar')
plt.xticks([], plt.yticks([]))

plt.show()
```

Orijinal



Kenarlar



Yöntemi fotoğraf üzerinde deneme:

Fotoğrafı okuma:

```
yuz_img = cv.imread('kız cocugu.png',0)  
yuz_kenar_img = cv.Canny(yuz_img, 50, 100)
```

Ekrana basma:

```
plt.figure(figsize=(12,16))  
  
plt.subplot(121), plt.imshow(yuz_img, cmap='gray'), plt.title('Orijinal')  
plt.xticks([], plt.yticks([]))  
  
plt.subplot(122), plt.imshow(yuz_kenar_img, cmap='gray'), plt.title('Kenar')  
plt.xticks([], plt.yticks([]))  
  
plt.show()
```



Bir tomografi görüntüsündeki kanser hücrelerini bu yöntemle tespit edilebilir.

KAYNAKÇA

Bilgeiř “Herkes için Yapay Zekâ II” eğitimi.