

GÖRÜNTÜ YUMUŞATMA

Yumuşatma, diğer adı ile bulanıklaştırma, görüntüdeki detayları azaltır. Bu sayede daha az görünümeler elde edilebilir. Akıllı telefonlardaki görüntü iyileştirme uygulamalarında cilt kusurlarını ve kırışıkları gideren filtre, bir yumuşatma filtresidir. Gereksiz detayları yok ederek daha bütüncül bir görüntü elde etmemizi sağlar. Fazla detaylar kaybolduğu için ayırt etmek istediğimiz daha büyük özelliklerin pozisyonlarını algılamak da kolaylaşır.

Yumuşatma amacı ile Ortalama Filtre ya da Gaussian Blur Filtre kullanabiliriz.

Ortalama filtre ile seçilen çekirdek kare, İngilizce terimi ile Kernel, içerisindeki tüm sayıların ortalamasını alarak tüm çekirdeği aynı renk değerine eşitler. Filtremiz ne kadar büyük olursa bulanıklaştırma işlemi o kadar yoğun olacaktır.

	104	100	108	
	99	106	98	
	95	90	85	

Örneğin; 3'e 3'lük bir çekirdek seçelim. İçerisinde toplamda 9 değer var. Bu nedenle bu çekirdek ile her bir işlem yaptığımız piksel değerinin 9'a bölerek birbirleriyle topluyoruz. Toplam ise bize çekirdeğin denk geldiği merkez için yeni değerini veriyor.

104	100	108
99	106	98
95	90	85

X

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

toplam= $104/9 + 100/9 + 108/9 + 99/9 + 106/9 + 98/9 + 95/9 + 90/9 + 85/9$
= 98.333

Uygulama

Drive bağlanma ve kütüphaneleri ekleme:

```
[1] from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

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

Resmimizi okuma:

```
resim1=cv2.imread('/content/gdrive/MyDrive/kodluyoruz/filter.jpg')
```

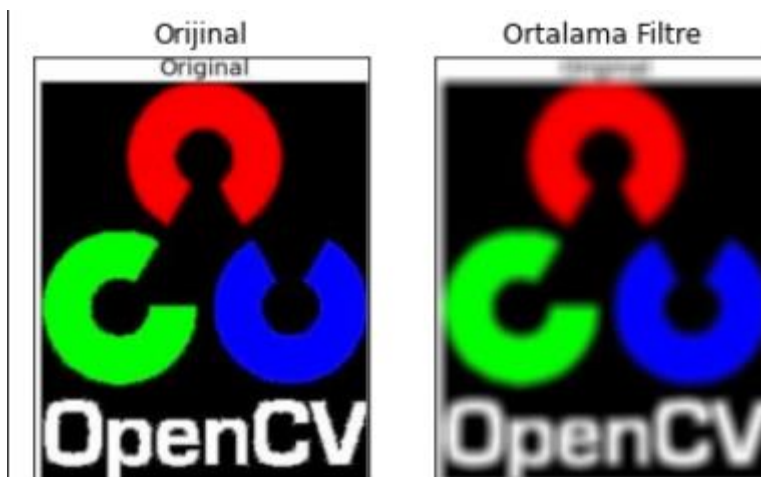
Ortalama filtre kullanmak için öncelikle bir çekirdek, yani Kernel'a ihtiyacımız var. 9x9'lik kernel oluşturduk ve çekirdek değerimize eşitledik. Küçük bir hatırlatma; seçtiğimiz kernel'ların boyutu tek sayı olmalı. Çünkü çift sayılı matrislerin merkez noktası olmaz.

```
cekirdek = np.ones((9,9),np.float32)/81
resim2 = cv2.filter2D(resim1,-1,cekirdek)

plt.subplot(121), plt.imshow(resim1), plt.title('Original')
plt.xticks([], plt.yticks([]))

plt.subplot(122), plt.imshow(resim2), plt.title('Ortalama Filtre')
plt.xticks([], plt.yticks([]))

plt.show()
```



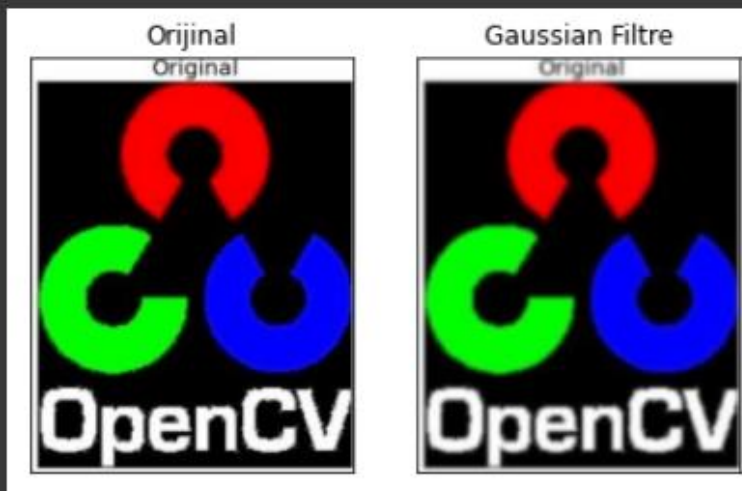
Yumuşatma amacı ile kullandığımız bir diğer filtremiz de Gauss Düzleştirme Filtresi.

```
resim3 = cv2.GaussianBlur(resim1,(5,5),0)

plt.subplot(121), plt.imshow(resim1), plt.title('Original')
plt.xticks([], plt.yticks([]))

plt.subplot(122), plt.imshow(resim3), plt.title('Gaussian Filtre')
plt.xticks([], plt.yticks([]))

plt.show()
```



Yüz kırışıklığında etkisini görme:

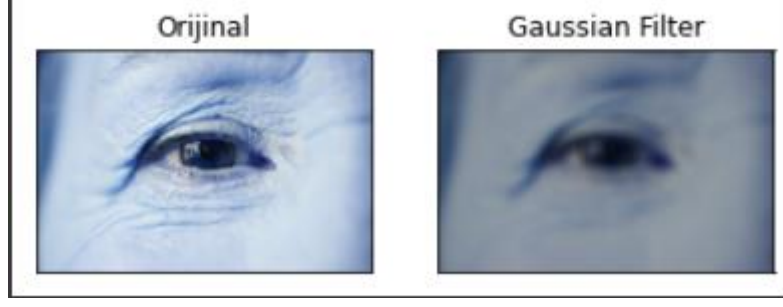
```
[9] resim4 = cv2.imread('/content/gdrive/MyDrive/kodluyoruz/eye.jpg')

[10] resim5 = cv2.GaussianBlur(resim4,(501,501),0)

plt.subplot(121), plt.imshow(resim4), plt.title('Original')
plt.xticks([], plt.yticks([]))

plt.subplot(122), plt.imshow(resim5), plt.title('Gaussian Filter')
plt.xticks([], plt.yticks([]))

plt.show()
```



Çizgilerin belirginliği azaldı. Filtreli halde daha düz bir cilt görüyoruz. Bu sırada göz bebeği ve iris e bir miktar bulanıklaştı.

KAYNAKÇA

Bilgeiř “Herkes için Yapay Zekâ II” eğitimi.

KODLUYORUZ
geleceęi kodluyoruz >_

 **EMpower**
Enriching young lives in emerging markets