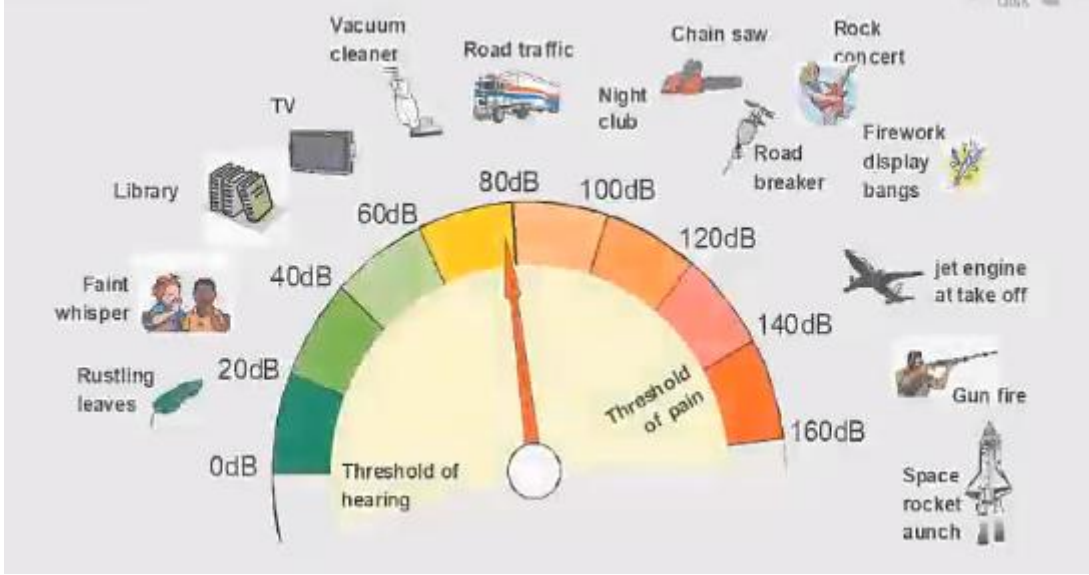


GÜRÜLTÜ GİDERME

Gürültü bir sinyal bir kanal üzerinde aktarılırken, çevredeki etkenler nedeni ile oluşan istenmeyen değişimlerdir.



Elektronik gürültü kavramına günlük hayattan bir örnek olarak telefon konuşmalarındaki cızırtı seslerini verebiliriz. Görüntülerde ise dağınık olarak bulunan anlamsız olarak karşımıza çıkmaktadır. Telefondaki cızırtı nasıl ses kalitesini bozuyorsa görüntülerdeki bu noktalar da görüntü kalitesini bozarlar.

Siyah beyaz resimde siyah ya da beyaz renkli olurken, renkli resimde renkli noktalar halinde bulunabilir. Fakat en sık karşılaşılanı beyaz noktacıklardır. Görüntülerdeki bu noktaların kaldırılması görüntünün iyileştirilmesini sağlar. Görüntüdeki gürültü çok sık değilse yumuşatma işlemi gürültüyü giderebilir. Fakat yoğun gürültüler için en etkin sonucu veren filtrelerden ikisi medyan ve bilateral filtrelerdir. Bunlar dışında da çeşitli gürültü giderme filtreleri bulunmaktadır.

Median

Bilateral

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:

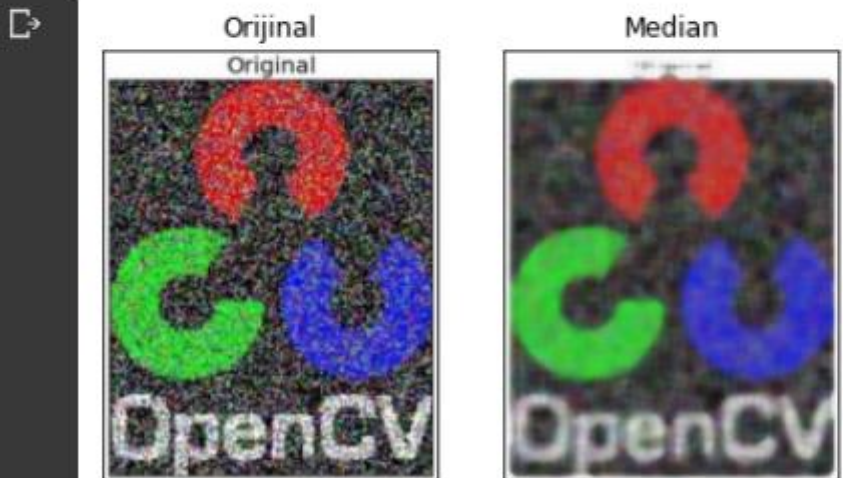
```
resim1=cv2.imread('/content/gdrive/MyDrive/kodluyoruz/median.jpg')
```

```
[4] resim2 = cv2.medianBlur(resim1,9)
```

```
plt.subplot(1,2,1), plt.imshow(resim1), plt.title('Orijinal')  
plt.xticks([]), plt.yticks([])
```

```
plt.subplot(1,2,2), plt.imshow(resim2), plt.title('Median')  
plt.xticks([]), plt.yticks([])
```

```
plt.show()
```



Filtreli resmimizde nokta yoğunlukları azaldı, OpenCV yazımız oldukça bulanıklaştı.

Şekillerin kenarlarında bu bulanıklaşmayı istemiyorsak bilateral filtre kullanabiliriz. 2 Yönlü bir filtre olan bilateral filtre cisimlerin sınırlarını korur ve sınırların iç ve dış bölümlerinde hem gürültüyü giderir hem de görüntünün yumuşamasını sağlar.

```
resim_bilateral = cv2.imread('/content/gdrive/MyDrive/kodluyoruz/bilateral.png')

[7] resim3 = cv2.bilateralFilter(resim_bilateral,9,100,100)

plt.subplot(1,2,1), plt.imshow(resim_bilateral), plt.title('Orijinal')
plt.xticks([], plt.yticks([]))

plt.subplot(1,2,2), plt.imshow(resim3), plt.title('Bilateral Filtered')
plt.xticks([], plt.yticks([]))

plt.show()
```

The image shows a side-by-side comparison of two grayscale photographs of a tomato. The left image, titled 'Orijinal', shows the tomato with some visible noise and less sharp edges. The right image, titled 'Bilateral Filtered', shows the same tomato but with significantly reduced noise and sharper, more defined edges, while maintaining the overall texture and color of the original image.

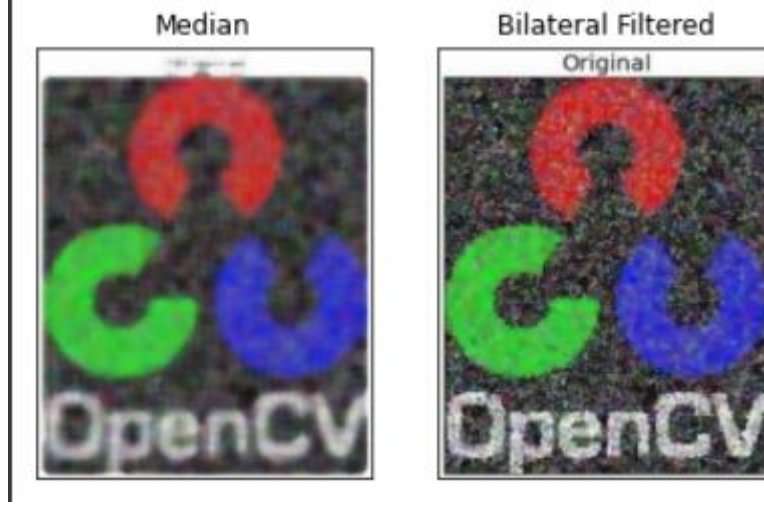
Resim1'in median ve bilateral halinin karşılaştırılması:

```
[9] resim4 = cv2.bilateralFilter(resim1,9,100,100)

plt.subplot(1,2,1), plt.imshow(resim2), plt.title('Median')
plt.xticks([], plt.yticks([]))

plt.subplot(1,2,2), plt.imshow(resim4), plt.title('Bilateral Filtered')
plt.xticks([], plt.yticks([]))

plt.show()
```



Median filtre tüm resmi bulanıklaştırdı. Bilateral filtrede ise yazılar ve şekil kenarları hala net.

KAYNAKÇA

Bilgeiş “Herkes için Yapay Zekâ II” eğitimi.

KODLUYORUZ
geleceği kodluyoruz >_

 **EMpower**
Enriching young lives in emerging markets