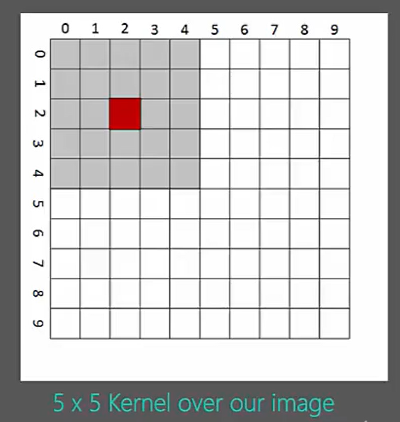
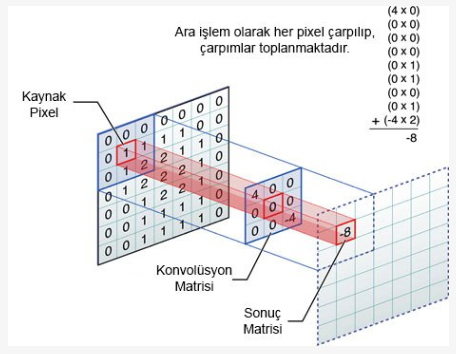
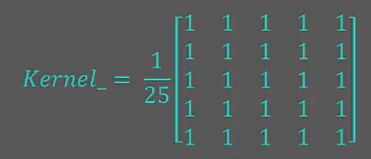
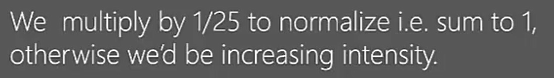
Convolutions and Blurring

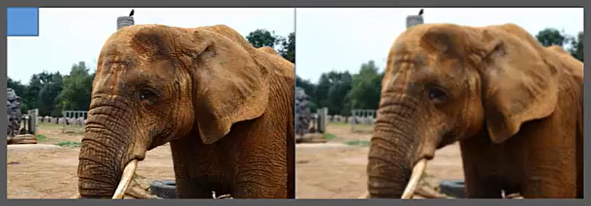












**ORTALAMA FİLTRESİ (Mean Filter -Box Blur)**

Yaygın isimleri; Mean filtering (ortalama filtresi), Smoothing (yumuşatma), Averaging (ortalama), Box filtering (kutu filtreleme).

Ortalama filtresi, görüntüleri yumuşatmanın basit ve uygulanması kolay bir yöntemidir. Diğer bir deyişle, bir piksel ile diğerleri arasındaki değişim miktarını azaltmaktır. **Genellikle görüntülerdeki gürültüyü azaltmak için kullanılır.**

Ortalama filtresi, bir görüntünün her bir piksel değerini komşularının ve kendisinin dahil olduğu ortalama değer ile değiştirmektir. Bu durum, çevresindekileri temsil etmeyen piksel değerlerinin ortadan kalkmasına yol açar.

Ortalama filtresi, bir görüntüdeki gürültüyü azaltmak için kullanılan en basit yöntemdir. Ancak gürültü daha az belirgin hale getirilirken, görüntüde yumuşatılmış olmaktadır. Kullanılan çekirdek şablonun (matrisin) boyutu artırılırsa yumuşatma daha da artacaktır.

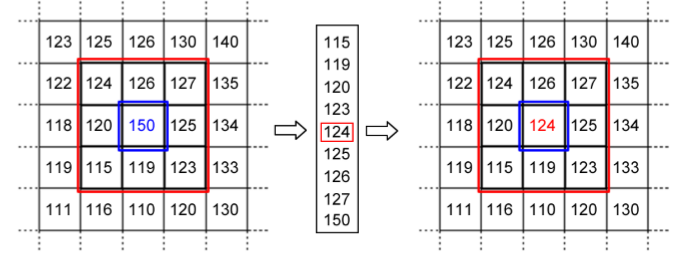
Ortalama filtrelemeyle ilgili iki ana sorun bulunmaktadır:

a) Resmi çok iyi temsil etmeyen değere sahip bir piksel, yakın bölgedeki tüm piksellerin ortalama değerini önemli ölçüde etkiler. Buda resmin değişmesine sebep olur.

b) Filtre (şablon) bir kenar üzerinden geçerken, kenarın her iki tarafındaki pikseller için yeni değerler üretecektir ve bu durum kenarın bulanıklaşmasına sebep olacaktır. Eğer keskin kenarların kaybolması istenmiyorsa bu bir sorun olabilir.

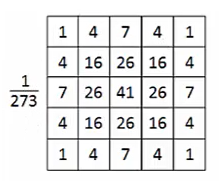
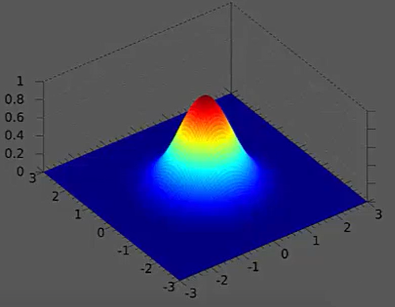
**MEDYAN FİLTRESİ (ORTA DEĞER FİLTRESİ-MEDİAN FİLTER)**

Bu filtrenin yaygın kullanılan isimleri; Medyan filtresi (Orta Değer filtresi) (Median filtering), Sıralama filtresi (Rank filtering)

Medyan filtresi, normal olarak mean filtresi gibi bir resimdeki gürültüyü azaltmak için kullanılır. Ancak resim üzerindeki detayların kaybolmaması noktasında mean filtresinden çok daha iyi sonuç verir.

Siyah ve beyazdan oluşan bir sınırda ortadaki değer ya siyah olur yada beyaz olur. İkisinin ortalaması olan gri olmayacaktır. Böylece kenar üzerindeki keskinlik kaybolmamış olacaktır.

**GAUSS BULANIKLAŞTIRMA/YUMUŞATMA FİLTRESİ (Gaussian Smoothing- Gaussian blur)**

****

**1 2 1**

**1/16 2 4 2**

**1 2 1**

**BİLETERAL FİLTRE**

Advanced Gauss.

Basic Thresholding Operations[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold#basic-thresholding-operations)

Goal[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold" \l "goal" \o "Permalink to this headline)

In this tutorial you will learn how to:

* Perform basic thresholding operations using OpenCV function [threshold](http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html?highlight=threshold#threshold)

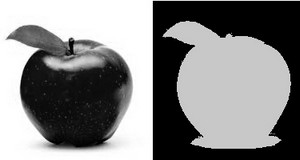
Cool Theory[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold" \l "cool-theory" \o "Permalink to this headline)

Note

The explanation below belongs to the book **Learning OpenCV** by Bradski and Kaehler.

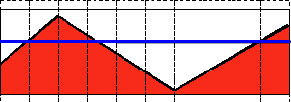
What is Thresholding?[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold#what-is-thresholding)

* The simplest segmentation method
* Application example: Separate out regions of an image corresponding to objects which we want to analyze. This separation is based on the variation of intensity between the object pixels and the background pixels.
* To differentiate the pixels we are interested in from the rest (which will eventually be rejected), we perform a comparison of each pixel intensity value with respect to a *threshold* (determined according to the problem to solve).
* Once we have separated properly the important pixels, we can set them with a determined value to identify them (i.e. we can assign them a value of 0(black), 255(white) or any value that suits your needs).



Types of Thresholding[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold" \l "types-of-thresholding" \o "Permalink to this headline)

* OpenCV offers the function [threshold](http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html?highlight=threshold#threshold) to perform thresholding operations.
* We can effectuate 5types of Thresholding operations with this function. We will explain them in the following subsections.
* To illustrate how these thresholding processes work, let’s consider that we have a source image with pixels with intensity values src(x,y). The plot below depicts this. The horizontal blue line represents the threshold thresh(fixed).

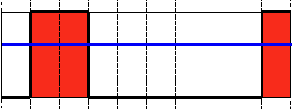


Threshold Binary[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold" \l "threshold-binary" \o "Permalink to this headline)

* This thresholding operation can be expressed as:

\texttt{dst} (x,y) =  \fork{\texttt{maxVal}}{if $\texttt{src}(x,y) > \texttt{thresh}$}{0}{otherwise}

* So, if the intensity of the pixel src(x,y)is higher than thresh, then the new pixel intensity is set to a MaxVal. Otherwise, the pixels are set to 0.

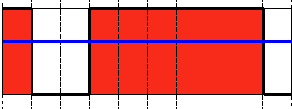


Threshold Binary, Inverted[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold" \l "threshold-binary-inverted" \o "Permalink to this headline)

* This thresholding operation can be expressed as:

\texttt{dst} (x,y) =  \fork{0}{if $\texttt{src}(x,y) > \texttt{thresh}$}{\texttt{maxVal}}{otherwise}

* If the intensity of the pixel src(x,y)is higher than thresh, then the new pixel intensity is set to a 0. Otherwise, it is set to MaxVal.

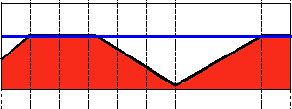


Truncate[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold" \l "truncate" \o "Permalink to this headline)

* This thresholding operation can be expressed as:

\texttt{dst} (x,y) =  \fork{\texttt{threshold}}{if $\texttt{src}(x,y) > \texttt{thresh}$}{\texttt{src}(x,y)}{otherwise}

* The maximum intensity value for the pixels is thresh, if src(x,y)is greater, then its value is *truncated*. See figure below:

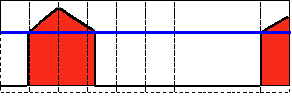


Threshold to Zero[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold#threshold-to-zero)

* This operation can be expressed as:

\texttt{dst} (x,y) =  \fork{\texttt{src}(x,y)}{if $\texttt{src}(x,y) > \texttt{thresh}$}{0}{otherwise}

* If src(x,y)is lower than thresh, the new pixel value will be set to 0.



Threshold to Zero, Inverted[¶](https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html?highlight=threshold" \l "threshold-to-zero-inverted" \o "Permalink to this headline)

* This operation can be expressed as:

\texttt{dst} (x,y) =  \fork{0}{if $\texttt{src}(x,y) > \texttt{thresh}$}{\texttt{src}(x,y)}{otherwise}

* If src(x,y)is greater than thresh, the new pixel value will be set to 0.

