



**NECMETTİN ERBAKAN
ÜNİVERSİTESİ**

**Mühendislik ve Mimarlık Fakültesi
Bilgisayar Mühendisliği
Görüntü İşlemeye Giriş Dersi Final Ödevi**

Öğrenci Bilgileri

Kader ARSLAN

16010011039

**Profesör Doktor
Sabri KOÇER**

**Haziran 2020
Konya**

Soru 1 : 1024x1024 boyutlarında herhangi bir gri seviye (8-bit) görüntüyü 128x128, piksel boyutlarında örnekleyerek resmin çözünürlüğünü azaltın. Bu resmin Negatifini Elde Etme ve transparent yapma işlemini. MATLAB kodlarını kullanarak yazınız ve görüntüleyiniz?

Öncelikle soruda yapacağımız işlemlerin Görüntü işlemedeki anlamlarına ve yapılanlara bakalım.

Örnekleme

- Görüntü elde etmenin pek çok yolu olduğunu gördük.
- Asıl amaç elde edilen verilerden görüntü elde etmektir.
- Çoğu sensörün çıkışı, genliği ve uzaysal davranışı görüntülenen olguya bağlı değişen sürekli voltaj dalgasıdır.
- Sayısal görüntü elde etmek için bu sürekli veriyi sayısal forma çevirmek gerekir.
- Aşağıdaki işlem gerekli:
 - Örneklem (Sampling)
- Elde edilen imge $f(x,y)$ hem x,y koordinatlarında hem de genlikte sürekli olabilir
- Dijital imge için ikisini de ayrık yapmak gerekir.
- Örneklem: Koordinat değerlerini sayısallaştırmadır.

Alt Örnek Elde Etme

Subsampling (Seyreltme, Altörneklem)

- Alt örneklem yaparak spatial çözünürlüğü azaltabiliriz.

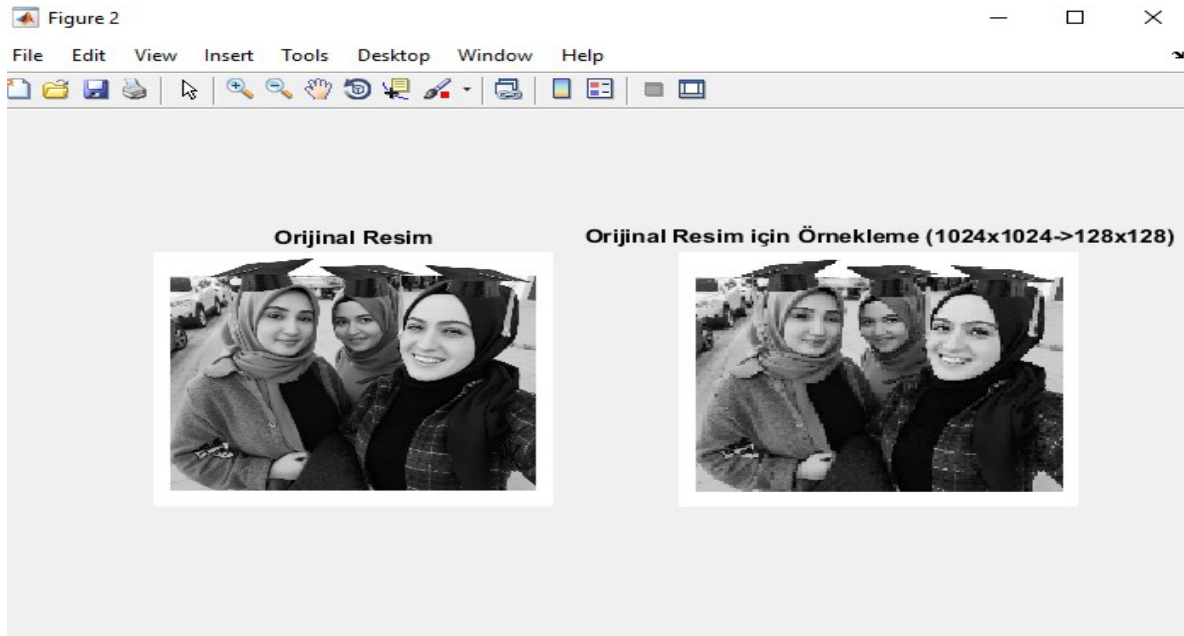
Uzaysal Çözünürlük

- Sayısal imgede örneklem yapılarak imgenin çözünürlüğü değiştirilebilir.

Çözünürlük imgenin pixel sayısı / birim fiziksel boyut ifadesi ile bulunur.

- Birimi ppi (pixels per inch) ya da dpi (dots per inch)'dir.
- Bir nesnenin bir bölümü ne kadar çok pixel ile ifade edilirse uzamsal çözünürlük o kadar yüksek olur.
- Ancak saklamak için gereken yerin de daha fazla olacağını unutmayalım.

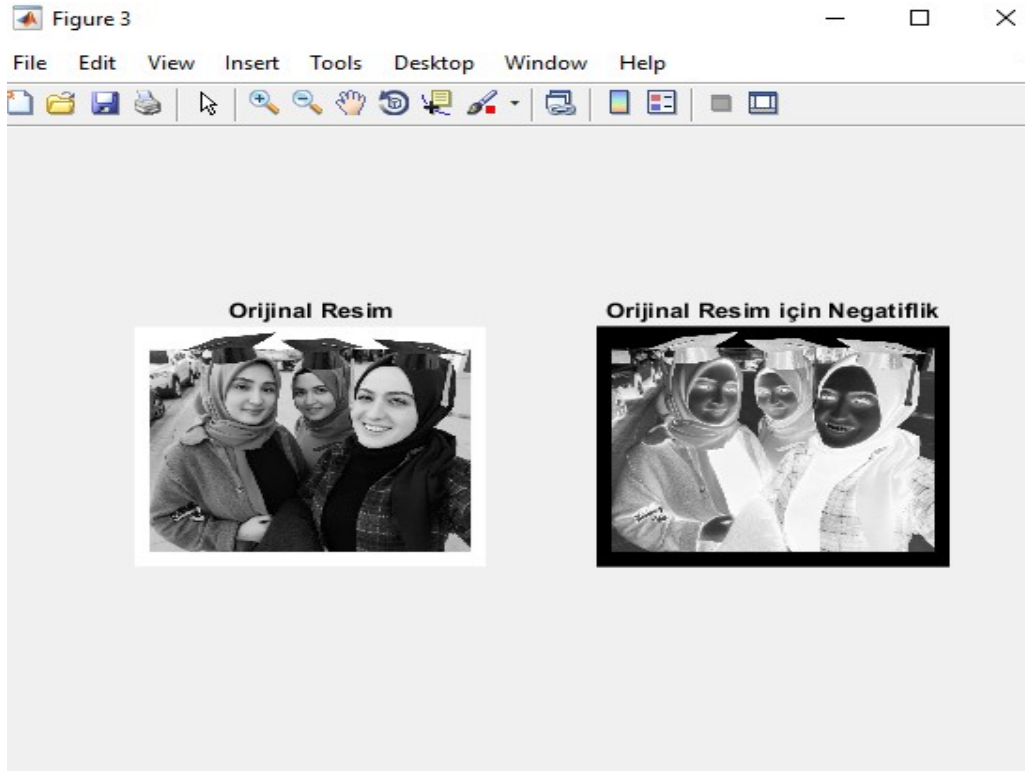
***Mezuniyetimiz olmadığı için en azından mezuniyet albümü yapmaya çalıştım:))**



Gri Seviyeli İmgeler İçin Temel Dönüşümler

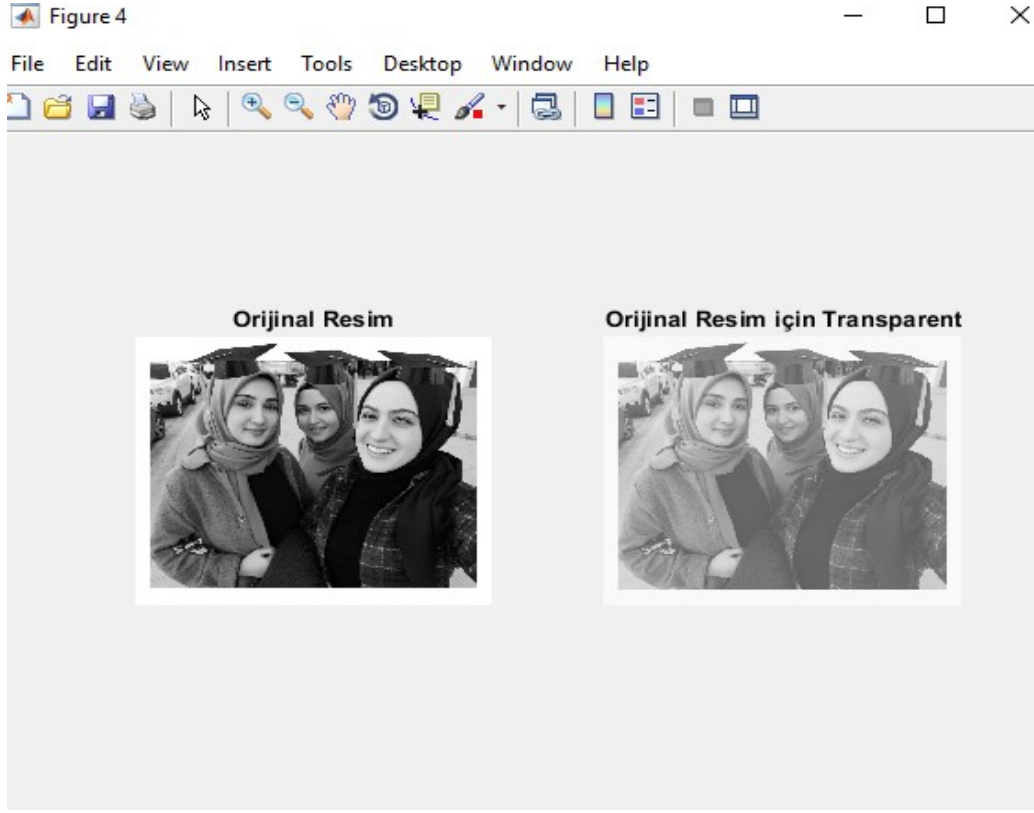
Görüntü Negatifleri

- $[0, L-1]$ aralığındaki bir imgenin negatifi için aşağıdaki fonksiyon kullanılır:
$$s = L - 1 - r$$
- Yapılan işlemler:
 - Yoğunluk seviyeleri tersine çevrilir
 - Beyaz ya da siyah ayrıntıların iyileştirilmesi gerektiğinde uygundur

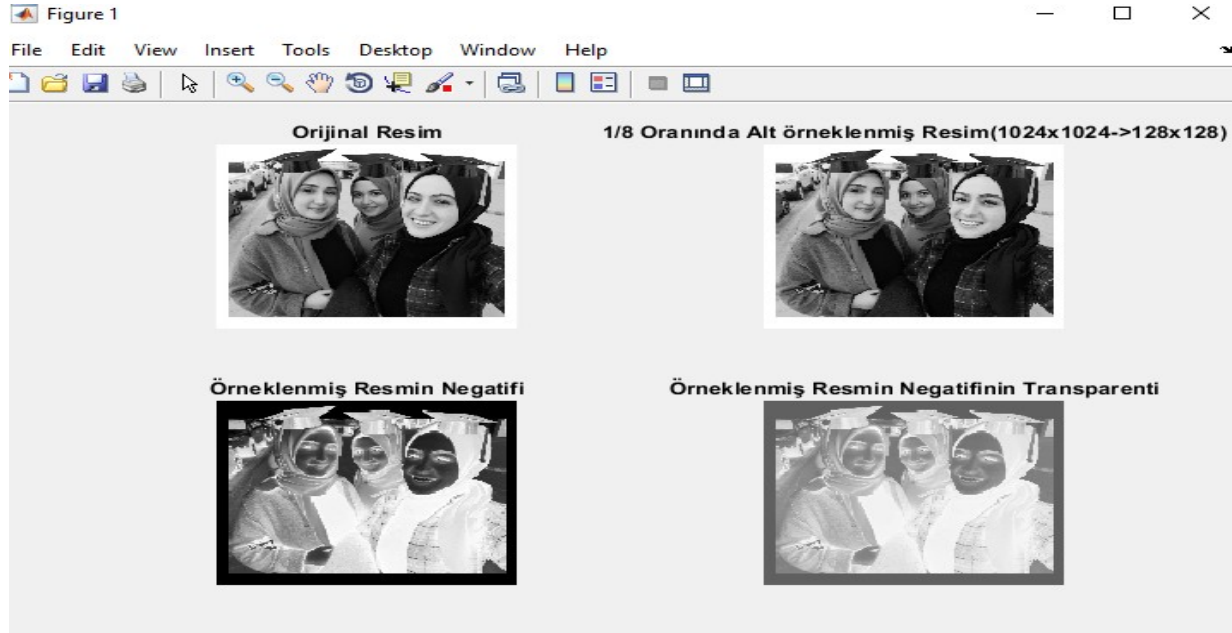


Transparent:

- Eksenlerdeki nesnelere saydamlık ekleme işlemidir.
- Bir grafik nesnesinin şeffaflığı, içinden görebileceğiniz dereceyi belirler. Grafiklerinizin görünümünü özelleştirmek için grafik nesnelere saydamlık ekleyin veya başka şekilde gizlenen bir nesne hakkındaki ayrıntıları gösterin. Bu tablo opak ve yarı saydam bir yüzey arasındaki farkı gösterir.
- $[0,1]$ arasındaki değerleri alır. 0 değeri tamamen saydam, 1 değeri tamamen opak ve 0 ile 1 arasındaki değerler yarı saydamdır.



Hepsinin birbirinin üstüne uygulanmış hali aşağıdaki görseldedir.



Yaptığım işlemlerin matlab kodu aşağıdadır:

```
img = imread('C:\Users\Casper\Desktop\images\mezun.png');  
img=rgb2gray(img); %resmin gri hali  
img1024=img(1:8:end,1:8:end); %resmin 1/8 oranında örnekleme  
img2=255-img1024; %negatifi
```

```
figure  
subplot(2,2,1);  
imshow(img);  
title('Orijinal Resim');
```

```
subplot(2,2,2), imshow(img1024), title('1/8 Oranında Alt örneklenmiş Resim(1024x1024->128x128)');
```

```
subplot(2,2,3), imshow(img2), title('Örneklenmiş Resmin Negatifi');
```

```
subplot(2,2,4);  
h=imshow(img2);  
set(h,'AlphaData',0.6) %transparent işlemi  
title('Örneklenmiş Resmin Negatifinin Transparenti');
```

```
figure  
subplot(1,2,1);  
imshow(img);  
title('Orijinal Resim');  
subplot(1,2,2);  
img1024=img(1:8:end,1:8:end);  
imshow(img1024);  
title('Orijinal Resim için Örnekleme (1024x1024->128x128)');
```

```
figure  
subplot(1,2,1);  
imshow(img);  
title('Orijinal Resim');  
subplot(1,2,2);  
img2=255-img;  
imshow(img2);  
title('Orijinal Resim için Negatiflik');
```

```
figure  
subplot(1,2,1);  
imshow(img);  
title('Orijinal Resim');  
subplot(1,2,2);  
t=imshow(img);  
set(t,'AlphaData',0.6);  
title('Orijinal Resim için Transparent');
```

Soru 2: Belirlediğiniz bir görüntü üzerinde uygun bir eşik değeri belirleyerek, Histogram Eşitleme ve Yoğunluk Dönüşüm Fonksiyonları uygulayınız. MATLAB kodlarını kullanarak yazınız ve görüntüleyiniz?

Öncelikle soruda yapacağımız işlemlerin Görüntü işlemedeki anlamlarına ve yapılanlara bakalım.

Histogram Eşitleme:

r : İyileştirilecek imgenin gri seviyeleri

$r \in [0 \ 1]$ 0-> siyah 1-> beyaz

pixel değerleri $\in [0 \ L-1]$

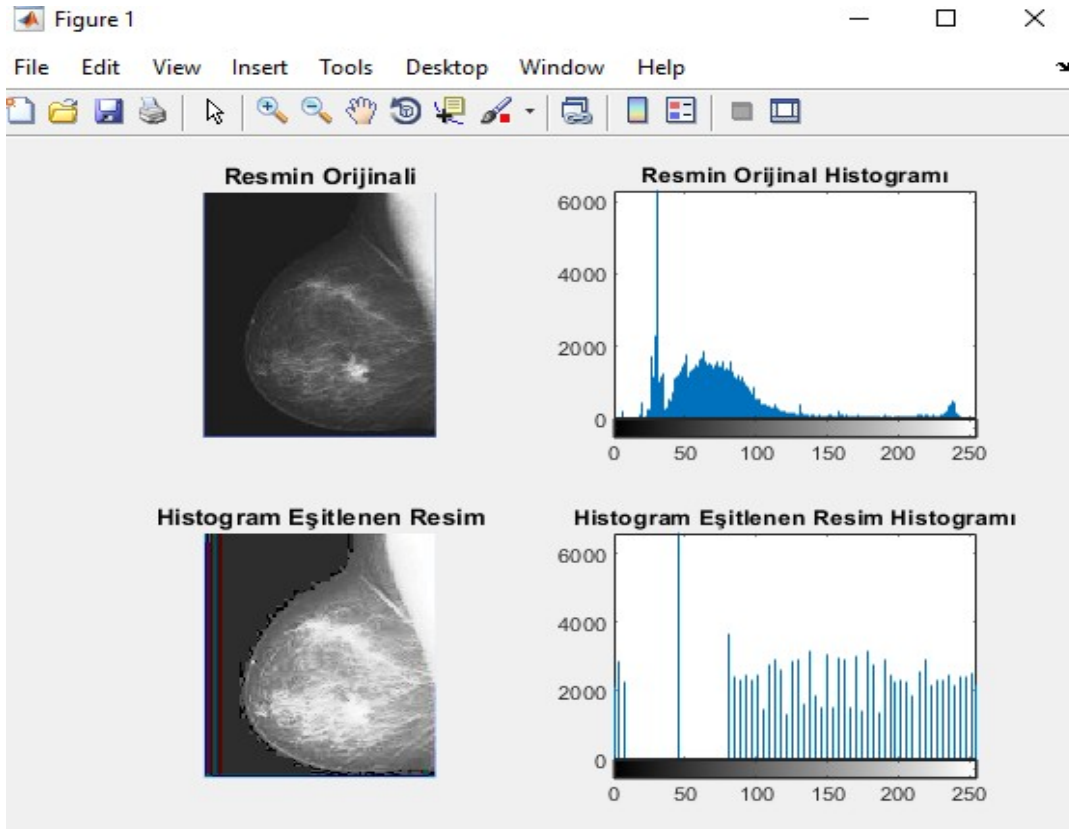
Dönüşüm

$$s = T(r), 0 \leq r \leq 1$$

Dönüşüm aşağıdaki şartları sağlamalı:

- $T(r)$ tek değerlidir
- Böylelikle bir geri dönüşümün varlığı garanti altına alınır.
- $0 \leq r \leq 1$ aralığında monotonik olarak artan bir fonksiyondur.
- Siyahtan beyaza yoğunluk artışı korunmuş olur.
- $0 \leq r \leq 1$ için $0 \leq T(r) \leq 1$
- Çıkış imgesi de giriş seviyeleri ile aynı olur.

Sonuç olarak elde edilen imge daha geniş bir gri seviyesi aralığını içerecektir.



Yoğunluk Dönüşüm Fonksiyonları:

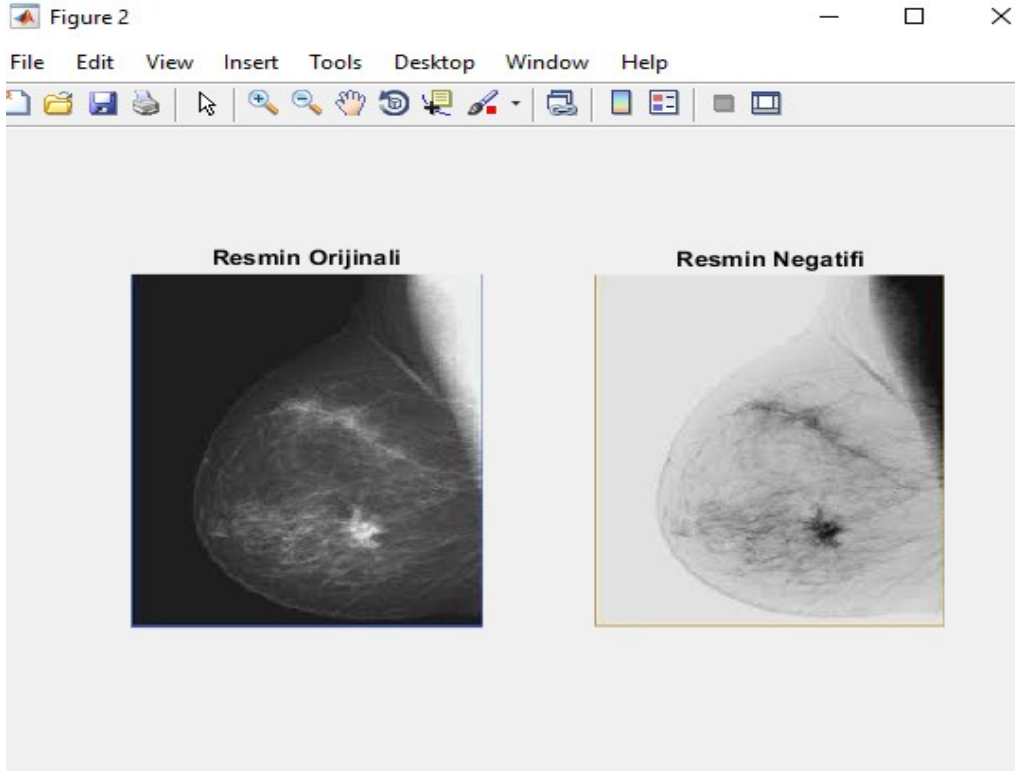
• Image Negatives

[0, L-1] aralığındaki bir imgenin negatifi için şekilde gösterilen fonksiyon kullanılır:

$$s = L - 1 - r$$

Yoğunluk seviyeleri tersine çevrilir.

Beyaz ya da siyah ayrıntıların iyileştirilmesi gerektiğinde uygundur.



• Log Transformations

Genel formu:

$$s = c \log (1 + r)$$

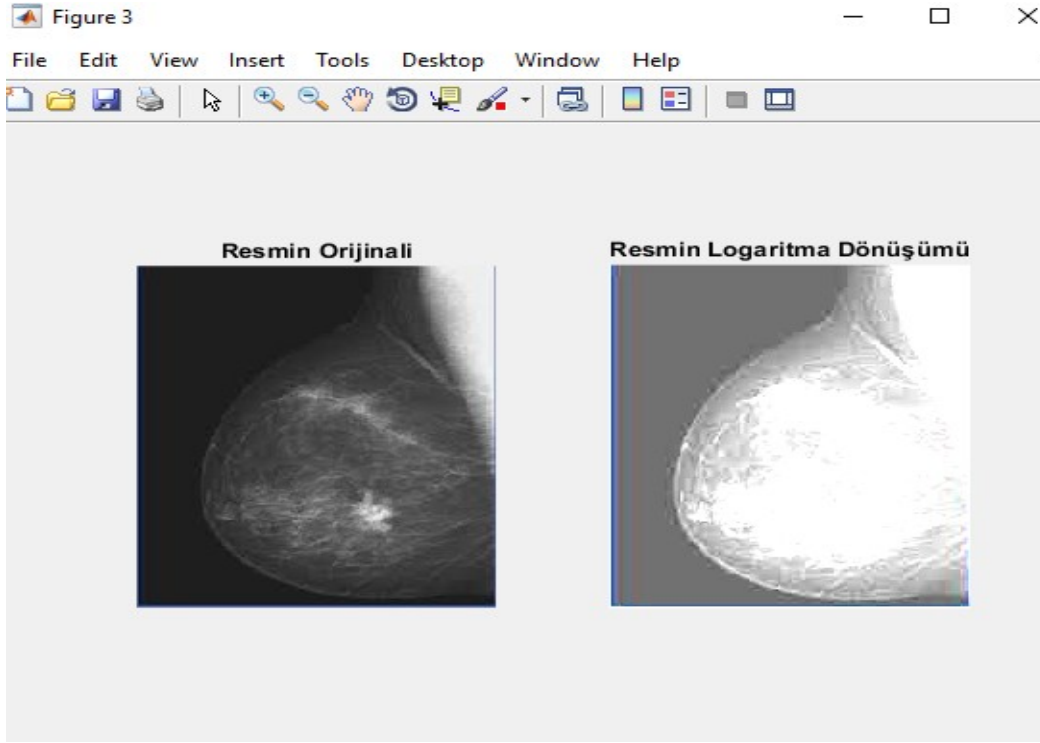
c: sabit, $r \geq 0$

Bu dönüşüm az aralıklı düşük gri seviyeli değerleri (siyaha yakın değerler) yüksek aralıklı hale getirir.

Yüksek seviyeli (beyaza yakın) değerler için tersi geçerli.

Koyu renkli pixelleri genişletmek için kullanılabilir.

Açık renklileri genişletmek için ters log dönüşüm kullanılır.



• Power-Law Transformations

Genel formu:

$s = cr^Y$ ya da input 0 olduğunda $c(r + \epsilon)^Y$

c ve Y sabit sayılar

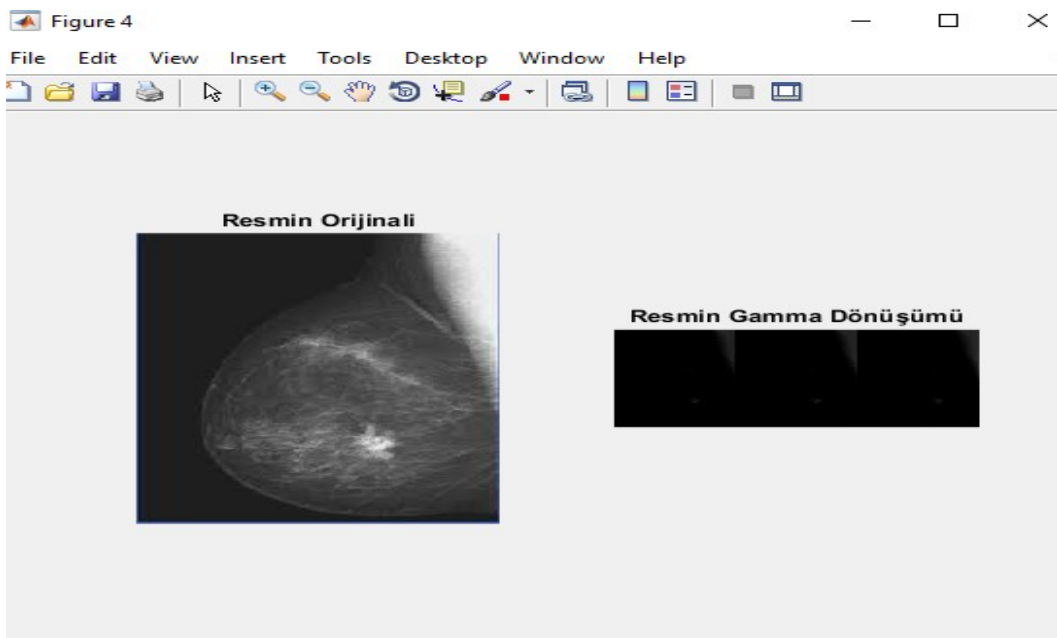
Gamma correction olarak da bilinir.

Piecewise-Linear Transformation Functions

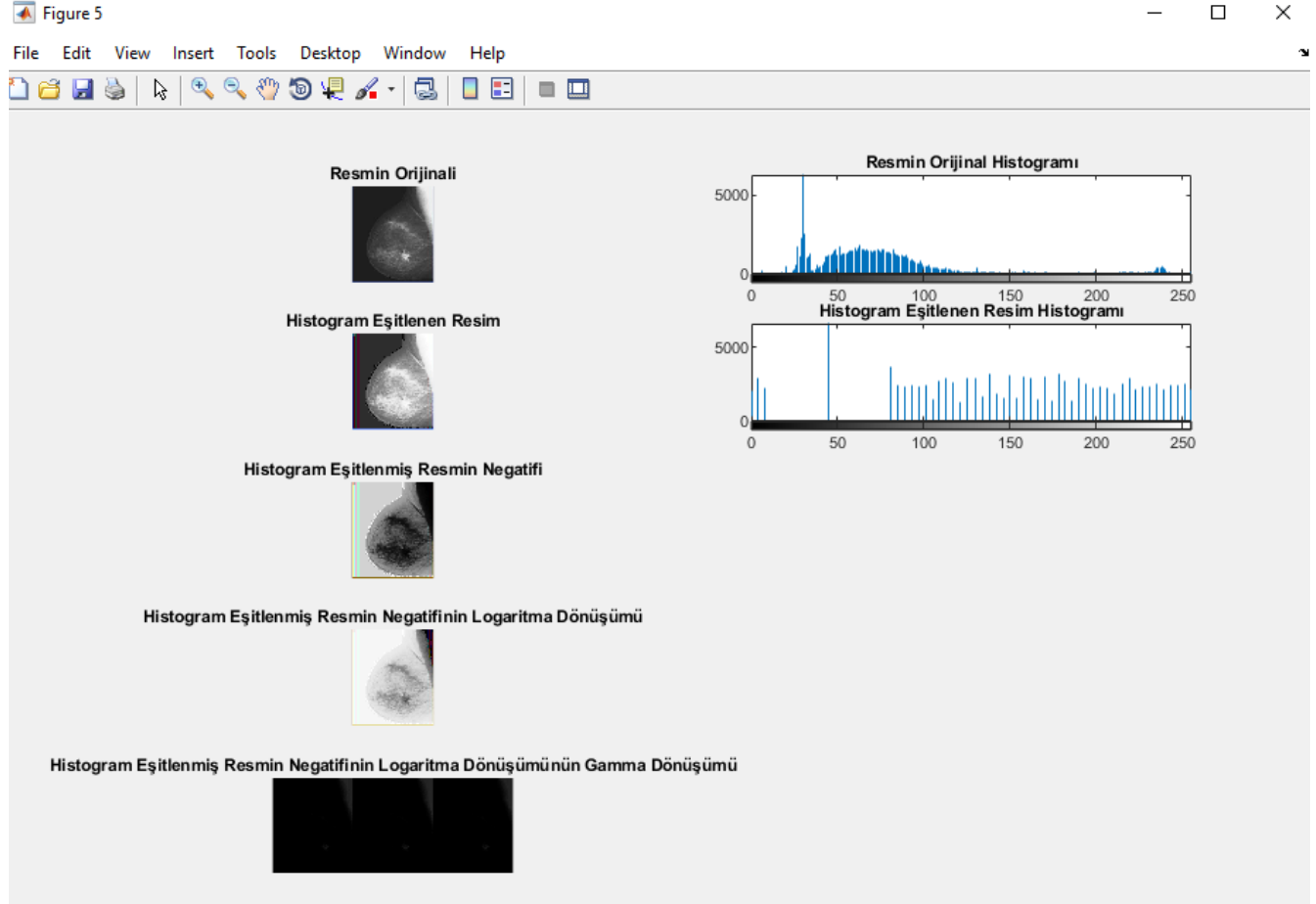
Log fonksiyonuna benzer. Ancak burada çok daha farklı curve elde edilebilir.

• Dar pencere kuyulu koyu renkli giriş değerlerini daha geniş bir çıkış aralığına haritalar.

$Y < 1$ ve $Y > 1$ durumları birbirinin tersi eğimler oluşturmaktadır.



Bütün istenilenlerin birbirini üzerine uygulanması aşağıdaki gibidir:



Yaptığım işlemlerin matlab kodu aşağıdadır:

```
close all;clc;clear all;
I = imread('C:\Users\Casper\Desktop\images\images.png');
I1=histeq(I);
adouble=im2double(I);
s=adouble;
[r,c]=size(adouble);
katsayilog=4;
katsayipower=0.20;
for i=1:r
    for j=1:c
        s(i,j)=katsayilog*log(1+adouble(i,j));
        y(i,j)=katsayipower*adouble(i,j)^5;
    end
end

img2=255-I;
figure
subplot(2,2,1), imshow(I), title('Resmin Orijinali');
```

```
subplot(2,2,2), imhist(I), title('Resmin Orijinal Histogramı');

subplot(2,2,3), imshow(I1), title('Histogram Eşitlenen Resim');
subplot(2,2,4), imhist(I1), title('Histogram Eşitlenen Resim Histogramı');

figure
subplot(1,2,1), imshow(I), title('Resmin Orijinali');
subplot(1,2,2), imshow(img2), title('Resmin Negatifi');
```

```
figure
subplot(1,2,1), imshow(adouble), title('Resmin Orijinali');
subplot(1,2,2), imshow(s), title('Resmin Logaritma Dönüşümü');
```

```
figure
subplot(1,2,1), imshow(adouble), title('Resmin Orijinali');
subplot(1,2,2), imshow(y), title('Resmin Gamma Dönüşümü');
```

%birbiri üzerine uygulama

```
figure
subplot(5,2,1), imshow(I), title('Resmin Orijinali');
subplot(5,2,2), imhist(I), title('Resmin Orijinal Histogramı');

subplot(5,2,3), imshow(I1), title('Histogram Eşitlenen Resim');
subplot(5,2,4), imhist(I1), title('Histogram Eşitlenen Resim Histogramı');

img2=255-I1;
subplot(5,2,5), imshow(img2), title('Histogram Eşitlenmiş Resmin Negatifi');

g=im2uint8(mat2gray(log(1+double(img2))));
subplot(5,2,7);
imshow(g);
title('Histogram Eşitlenmiş Resmin Negatifinin Logaritma Dönüşümü')

gama = imadjust(y, [], [], 5.0);
subplot(5,2,9), imshow(y), title('Histogram Eşitlenmiş Resmin Negatifinin Logaritma Dönüşümünün Gamma Dönüşümü');
```

Soru 3: Uzamsal süzme (spatial filters) methodlarından Sobel, Prewitt ve Laplace kenar bulma filtrelerini kullanarak herhangi bir resmin kenarlarını keskinleştiriniz. . MATLAB kodlarını kullanarak yazınız ve görüntüleyiniz?

Öncelikle soruda yapacağımız işlemlerin Görüntü işlemedeki anlamlarına ve yapılarına bakalım.

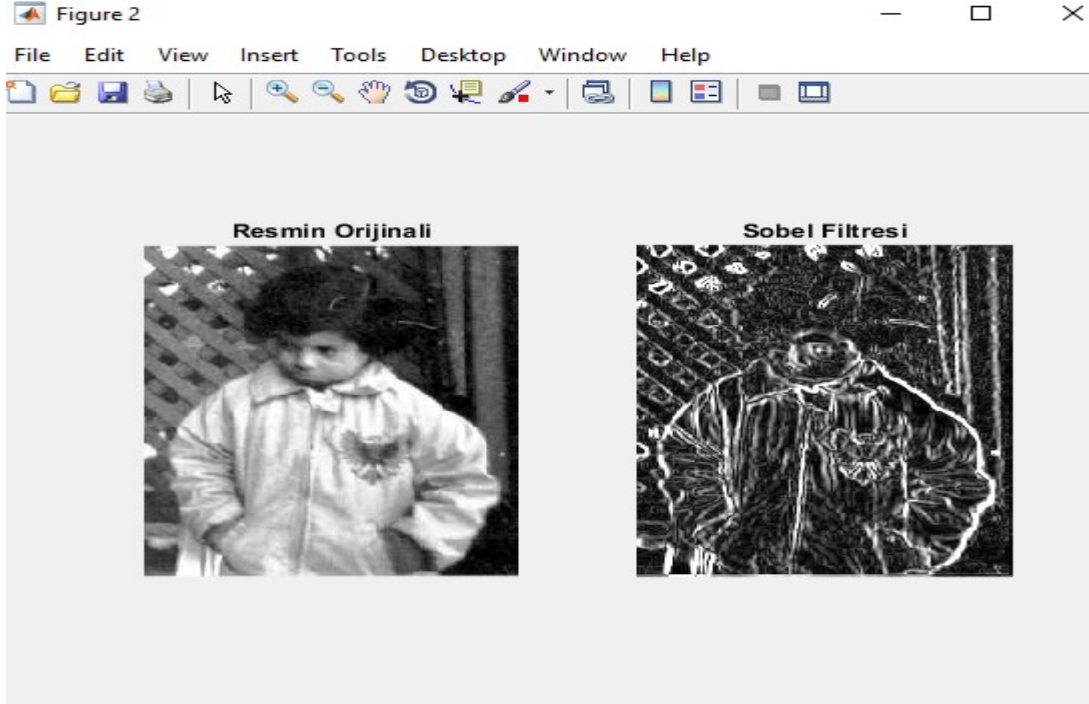
Uzamsal Süzme fonksiyonları:

Sobel:

Sobel kenar dedektörü, biri x-yönündeki (Gx maskesi) eğimi, diğeri y-yönündeki (Gy maskesi) eğimi (1. derece türev) hesaplamak üzere 3x3'lük bir konvolüsyon maske çifti kullanır. Bir maske diğeri 90° döndürülmüş halidir (saat yönü tersinde).

Mantık olarak bir kerede bir pikseli işleyerek maske görüntüde hareket eder.

Genellikle gri tonlamalı giriş görüntüsü üzerinde bulunan her bir noktadaki yaklaşık mutlak gradient (birinci türev-eğim) büyüklüğünü bulmak için kullanılır.

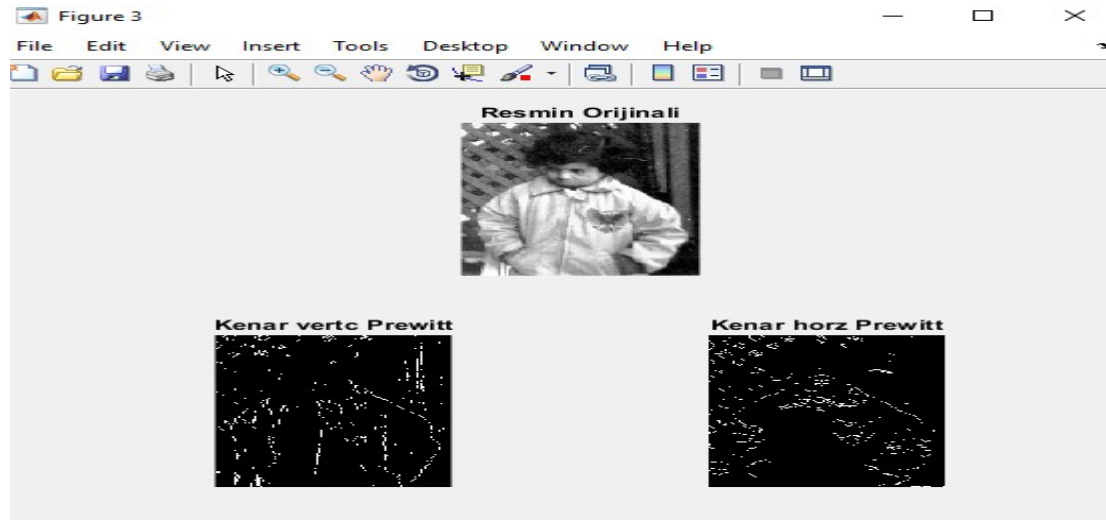


Prewitt:

Prewitt kenar yakalama yöntemi, birinci derece türevleri olan Gx ve Gy maskeleri kullanarak görüntüyü filtreler. Prewitt filtresi görüntü üzerinde filtre gezdirilmesi bakımından daha az zaman harcar.

Prewitt dedektör, sobel dedektöre göre hesaplama uygulamak için biraz daha basit, ama biraz noiser sonuçlar üretme eğilimindedir.

Matlab fonksiyonu olarak Sobel'in parametreleri ile aynıdır. Bir maske diğeri 90° döndürülmüş halidir (saat yönü tersinde). Kernel matrisi Sobel'den farklıdır.



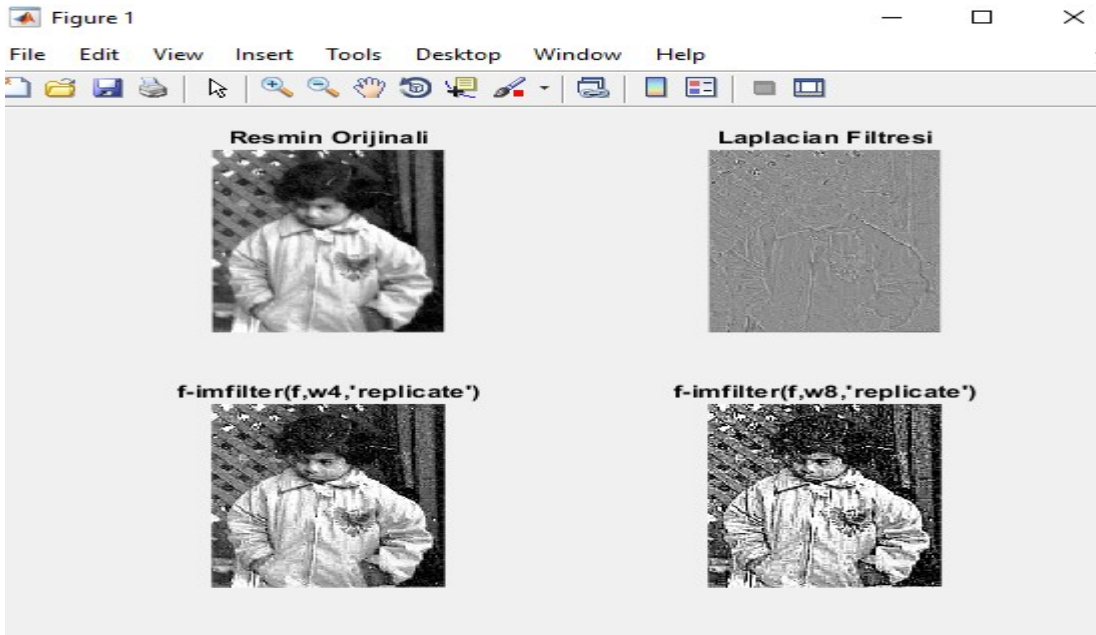
Laplace:

2-Boyutlu bir $f(x, y)$ fonksiyonunun x ve y deęişkenlerine göre 2.türevi, $f(x, y)$ 'nin laplasian olarak adlandırılır.

Laplasien kenar tanıma yönteminde, görüntü fonksiyonu $f(x,y)$ 'yi verilen laplasian kenar maskeleri ile konvolüsyona tabi tuttukten sonra elde edilen laplasian görüntüsünün mutlak deęerinin bir eşik deęeri ile karşılaştırılması sonucunda kenar görüntüsüne ulaşılır.

Süzgeçlenmiş (filtrelenmiş) görüntüdeki (laplasian görüntüsü) tüm sıfır geçiř noktaları (zero-crossing points) elde edilir.

Laplasian yönteminde, görüntüdeki en küçük gri seviyelerdeki deęişimlerde bile sıfır geçiř noktası üretebileceęi ve görüntünün küçük seviyelerde de olsa gürültü ile etkilenmiş olduęu göz önüne alınırsa, lablasian tabanlı yöntemlerin doğrudan doğruya görüntü üzerine uygulanması pek sık kullanılan bir yol deęildir. Bu durum aşağıdaki şekillerden de görölmektedir.



Yaptıęım işlemlerin matlab kodu aşağıdadır:

```
clc,clear all,close all;
```

```
A=imread('C:\Users\Casper\Desktop\images\his.png');
```

```

G=rgb2gray(A);
w4=fspecial('laplacian',0);
f=im2double(G);
g4=imfilter(f,w4,'replicate');
figure
subplot(2,2,1), imshow(G), title('Resmin Orijinali');
subplot(2,2,2), imshow(g4,[]), title('Laplacian Filtresi');
w4=fspecial('laplacian',0);
w8 = [1 1 1; 1 -8 1; 1 1 1];
f=im2double(G);
g4=f-imfilter(f,w4,'replicate');
g8=f-imfilter(f,w8,'replicate');
subplot(2,2,3), imshow(g4);title('f-imfilter(f,w4,"replicate")')
subplot(2,2,4), imshow(g8);title('f-imfilter(f,w8,"replicate")')

B=rgb2gray(A);
C=double(B);
for i=1:size(C,1)-2
    for j=1:size(C,2)-2
        % X yönü için sobel maskesi
        Gx=((2*C(i+1,j+1)+C(i,j+2)+C(i+2,j+2)-(2*C(i,j+1)+C(i,j)+C(i,j+2))));
        % Y yönü için sobel maskesi
        Gy=((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-(2*C(i+1,j)+C(i,j)+C(i+2,j)));
        % Görüntünün gradyanı
        B(i,j)=sqrt(Gx.^2+Gy.^2);
    end
end
figure
subplot(1,2,1), imshow(G); title('Resmin Orijinali');
subplot(1,2,2), imshow(B); title('Sobel Filtresi');

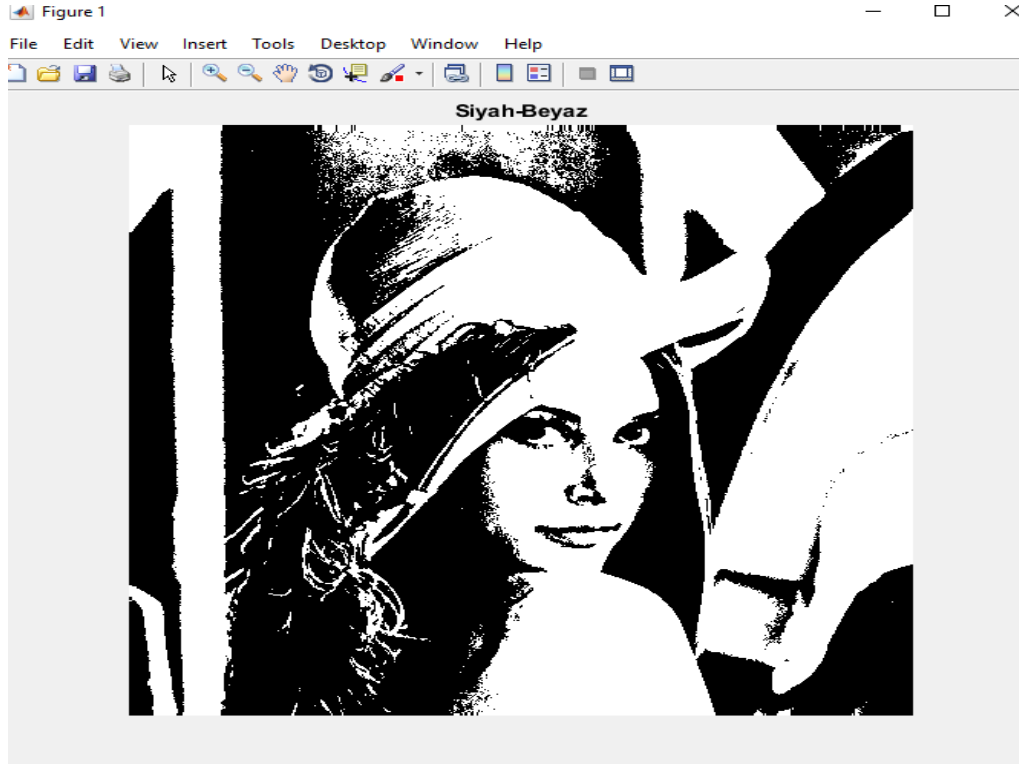
A=imread('C:\Users\Casper\Desktop\images\his.png');
g = rgb2gray(A);
s13=edge(g,'prewitt','vertical');
s14=edge(g,'prewitt','horizontal');
figure
subplot(2,2,1.5), imshow(g), title('Resmin Orijinali')
subplot(2,2,3), imshow(s13), title('Kenar vertc Prewitt')
subplot(2,2,4), imshow(s14), title('Kenar horz Prewitt')

```

Soru 4:Lena.tif resmini internetten bulup, Siyah Beyaz'a dönüştürünüz, 0,5 oranında boyutlandırma ve 60 derece döndürme ve yatay ters çevrilmiş (mirror) hallerini ayrı ayrı ekranda görüntüleyecek kodları MATLAB ortamında yazınız?

Öncelikle soruda yapacağımız işlemlerin Görüntü işlemedeki anlamlarına ve yapılanlara bakalım.

Siyah-Beyaz:



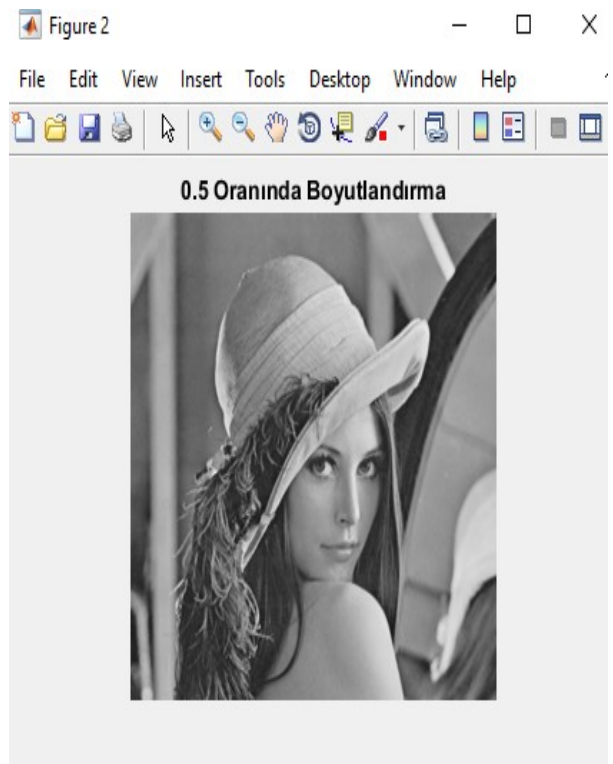
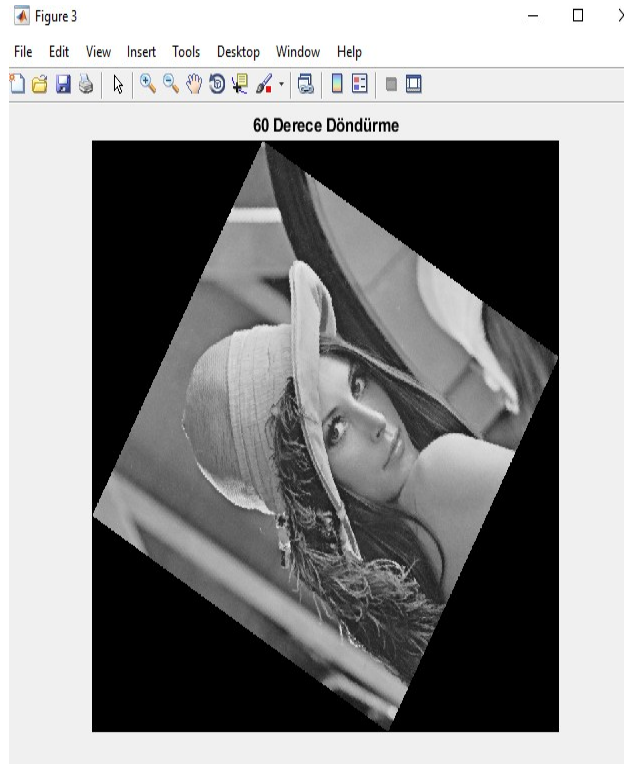
Döndürme & Boyutlandırma:

Döndürme, oranlı büyütme, x ve y eksenlerinde değişik oranlarda **boyutlandırma** işlemleri 2 aşama ile gerçekleştirilir.

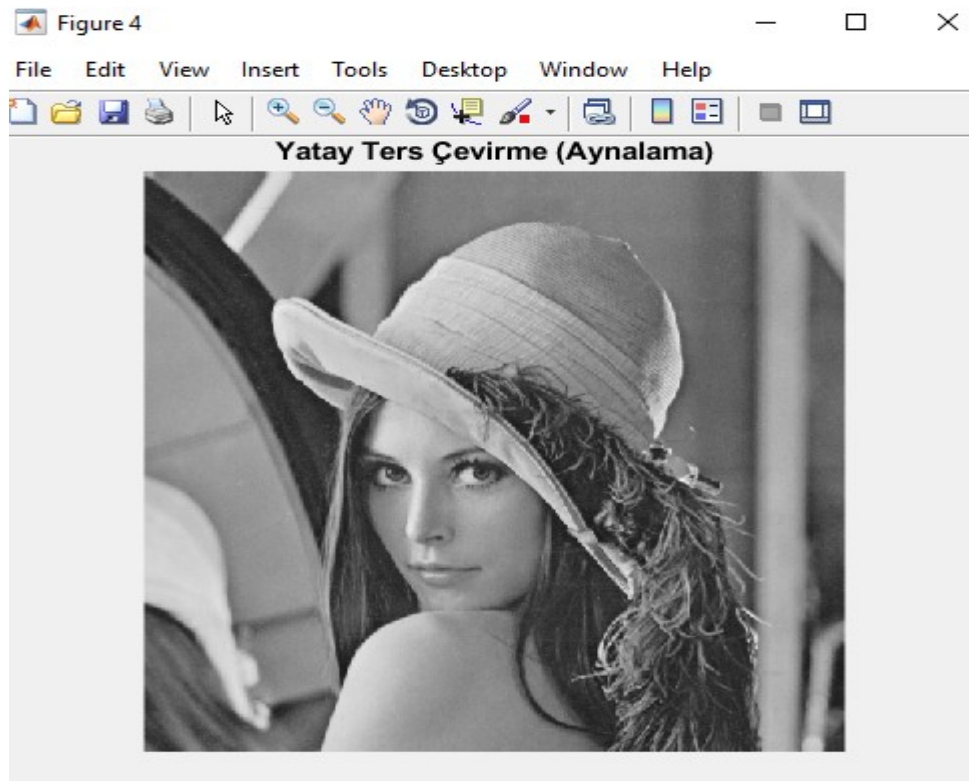
- Eldeki piksel değerleri kullanılarak gerekli dönüşümler yapılır.
- Entropolasyon kullanarak dönüşüme uğramış resmin tamsayı piksel koordinatları bulunur.

Entropolasyon: İlk kez Uygulamalı Matematik biliminin bir alt kategorisi olan Sayısal Analiz yöntemlerinde tanımlanan ve elde varolan (bilinen) değer noktalarından yola çıkarak bu noktalar arasında, farklı bir yerde ve değeri bilinmeyen bir noktadaki olası değeri bulmaya/tahmin etmeye yarayan yöntemlerin tümüne verilen genel isimdir.

İnterpolasyon genelde mühendislik ve deneylere/ölçümlere dayalı benzeri bilim dallarında, toplanan verilerin bir fonksiyon eğrisine uydurulması amacıyla kullanılmaktadır.



Yatay Ters Çevirme(Aynalama):



Yaptığım işlemlerin matlab kodu aşağıdadır:

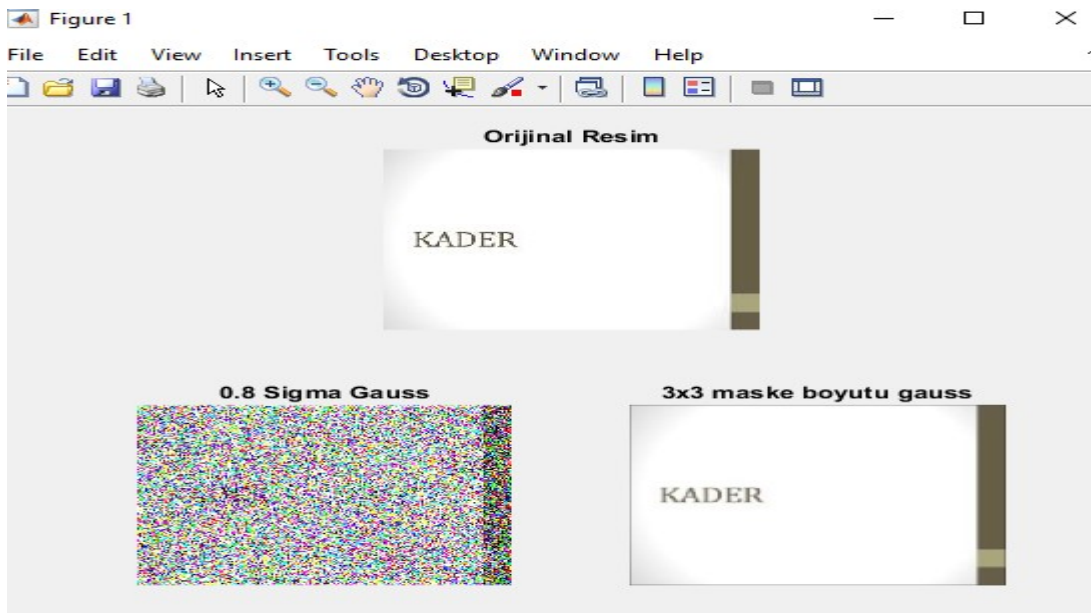
```
img=imread('C:\Users\Casper\Desktop\images\lena512color.tiff');
figure;
siyahbeyaz=rgb2gray(img);
A=im2bw(siyahbeyaz);
imshow(A);
title('Siyah-Beyaz');
figure;
boyutlandir=imresize(siyahbeyaz,0.5);
imshow(boyutlandir);
title('0.5 Oranında Boyutlandırma');
figure;
dondur=imrotate(siyahbeyaz,60,'nearest');
imshow(dondur);
title('60 Derece Döndürme');
figure;
mirror_image = flip(siyahbeyaz,2);
imshow(mirror_image);
title('Yatay Ters Çevirme (Aynalama)');
```

Soru 5:İnternettten bulacağınız isminizin baş harfiyle başlayan bir görüntünün ..

a. 0.8 sigma katsayılı 3*3 maske boyutlu gaussian filtrelenmiş hali

Gauss:

Gauss modeli doğal gürültü sürecini modellemek için kullanılan en genel yöntemdir. Bu yöntem özellikle görüntü elde etme sürecinde elektriksel olarak ortaya çıkan gürültüleri modellemek için kullanılır. Bu gürültü görüntüde gelişigzel dalgalanmalara sebep olur. Görüntü fonksiyonu $I(x,y)$ ise ve $N(x,y)$ 'de Gaussian gürültüsünü temsil ediyorsa; gürültülü görüntü; $g = I + N$ ile ifade edilir. Matlabda Gaussian gürültüsü eklemek için; $d = \text{imnoise}(g, 'gaussian', M, V)$ şeklindedir. Burada M ortalama, V variance değerleridir. Default olarak $M=0$ $V=0.01$ alınır.



Gauss matlab kodu aşağıdadır:

```
%0.8 sigma katsayılı 3x3 Gauss
g=imnoise(J1, 'gaussian',0,0.8);
filtre = fspecial('gaussian', [3 3], 0.8);
c = imfilter(J1,filtre); % filtre matrisi tüm imgeye işleniyor
figure
subplot(2,2,1.5), imshow(J1), title('Orijinal Resim');
subplot(2,2,3), imshow(g), title('0.8 Sigma Gauss');
subplot(2,2,4), imshow(c), title('3x3 maske boyutu gauss');
```

b. 5*5 maske boyutlu medyan filtrenmiş hali ve

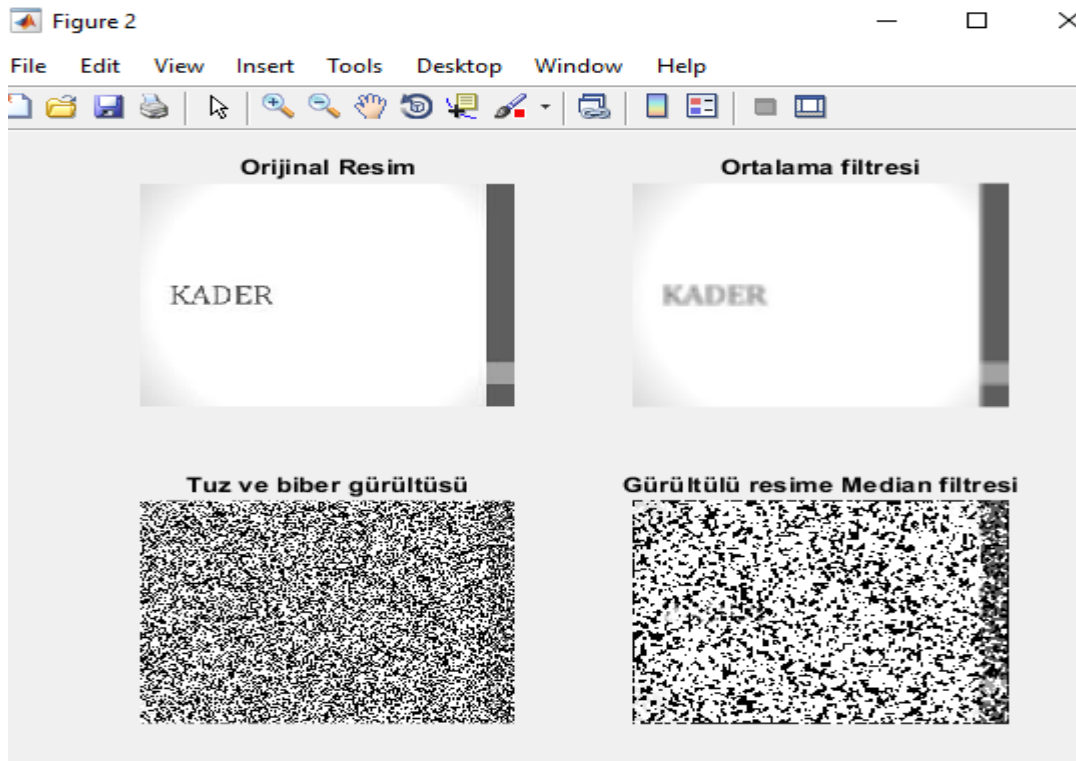
Medyan:

Median filtreleme, tuz-biber gürültüsünü yok etmek için çok uygundur. Medyan filtreler nonlineer uzaysal filtrelerdir. Maskeyi oluşturan boyuttaki resim piksel değerlerinin küçükten büyüğe sıralanıp ortadaki değeri merkez piksele atama işlemidir.

Bu filtrelerin bu tip gürültüler için, en az detay kaybı ile birlikte çok iyi çalıştığını görmekteyiz. Pencere boyu büyüdükçe bilgi kaybının oranı gittikçe artmaktadır.

Tuz & biber:

Tuz ve biber gürültüsü yüksek frekans bileşenli görüntüdür. Bunun için alçak geçiren filtreler kullanarak bu gürültü tipini yok edebiliriz veya azaltabiliriz. Bunun için bir ortalama filtre (Medyan filtreleri görüntü kenarlarını ve detaylarını bulanıklaştırma dezavantajına sahiptirler - Aslında Medyan filtreleri gauss ve uniform gürültü tiplerinde oldukça etkilidir.) deneyebiliriz. Yapılması gereken, fspecial('average') ile bir ortalama filtre matrisi oluşturup, bunu gürültülü görüntü ile konvoluse etmektir.



Medyan matlab kodu aşağıdadır:

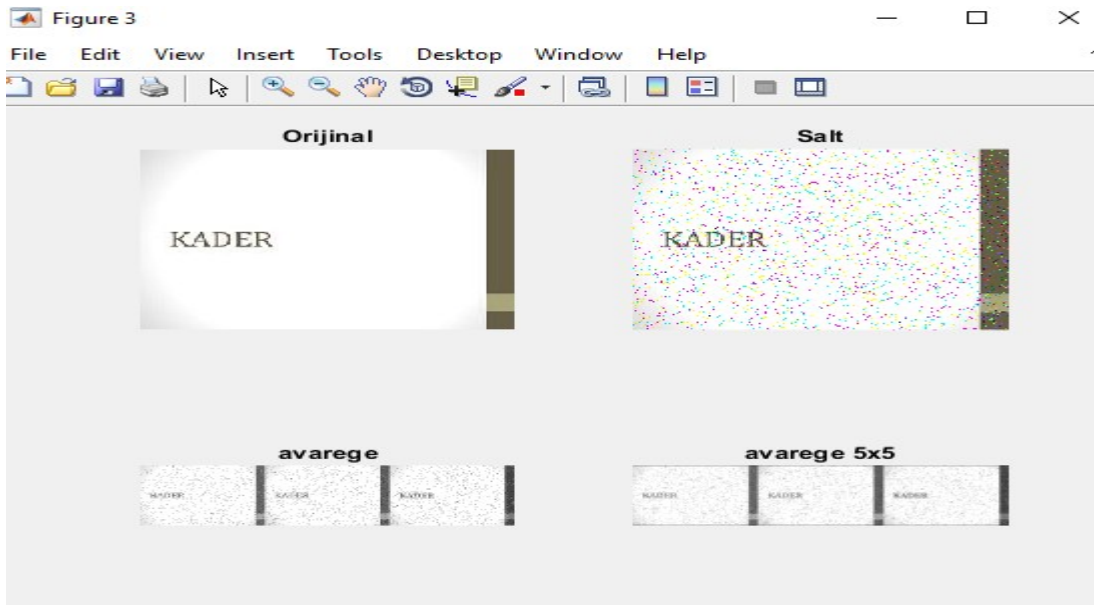
%5x5 median

```
imgGri = rgb2gray(J1);  
imgDou = im2double(imgGri);  
hOrt = fspecial('average',[5,5]);  
  
imgOrt = imfilter(imgDou,hOrt,'replicate');  
imgSap = imnoise(imgGri,'salt & pepper', 0.8);  
imgMed = medfilt2(imgSap);  
figure;  
subplot(2,2,1); imshow(imgGri); title('Orijinal Resim');  
subplot(2,2,2); imshow(imgOrt); title('Ortalama filtresi');  
subplot(2,2,3); imshow(imgSap); title('Tuz ve biber gürültüsü');  
subplot(2,2,4); imshow(imgMed); title('Gürültülü resime Median filtresi');
```

c. 5*5 maske boyutlu average filtrelenmiş halini matlab komutları kullanarak yazınız.

Average:

Mean filtreleri NxN penceresi içerisinde bir ortalama değer bulmaya çalışarak işlerler. Bunu yaparken görüntü üzerinde gezen bir kayan pencere yöntemi kullanırlar. Bu filtrelerin en basitlerinden birisi aritmetik mean filtresidir. Bu filtre pencere içerisindeki değerlerin aritmetik ortalamasını bulurlar. Aritmetik mean filtresi, görüntü içerisindeki yerel değişimleri yumuşatır yani aslında bu bir alçak geçiren filtredir. Bu, tüm katsayıları $1/N^2$ ile çarpılarak, maskenin görüntü üzerine konvolusyonu ile elde edilebilir. Bu filtre gürültü etkilerini yok ederken görüntüyü bulanık bir hale getirmeye çalışacaktır. Maske boyutları büyüdükçe bulanıklaşma etkileri o kadar çok meydana gelmektedir. Bu tip filtreler en iyi şekilde gauss ve uniform gürültülerde çalışırlar. Eğer gauss gürültüsünün ortalaması 0'sa gürültü filtreleme ile 0 yapılabilir. Ancak bir miktarda görüntüde bulanıklaşmayı göz önüne alınız. Filtre boyutu 3x3 veya 5x5 olabilir.



Average matlab kodu aşağıdadır:

```
%5x5 average
d=imnoise(J1,'salt & pepper');
t=im2double(d);
a=fspecial ('average');
b=fspecial ('average', [5,5]);
tt= filter2(a, t);
tt1= filter2(b, t);
figure
subplot(2,2,1),imshow(J1),title('Orijinal');
subplot(2,2,2),imshow(d),title('Salt');
subplot(2,2,3), imshow(tt),title('avarege');
subplot(2,2,4), imshow(tt1),title('avarege 5x5');
```