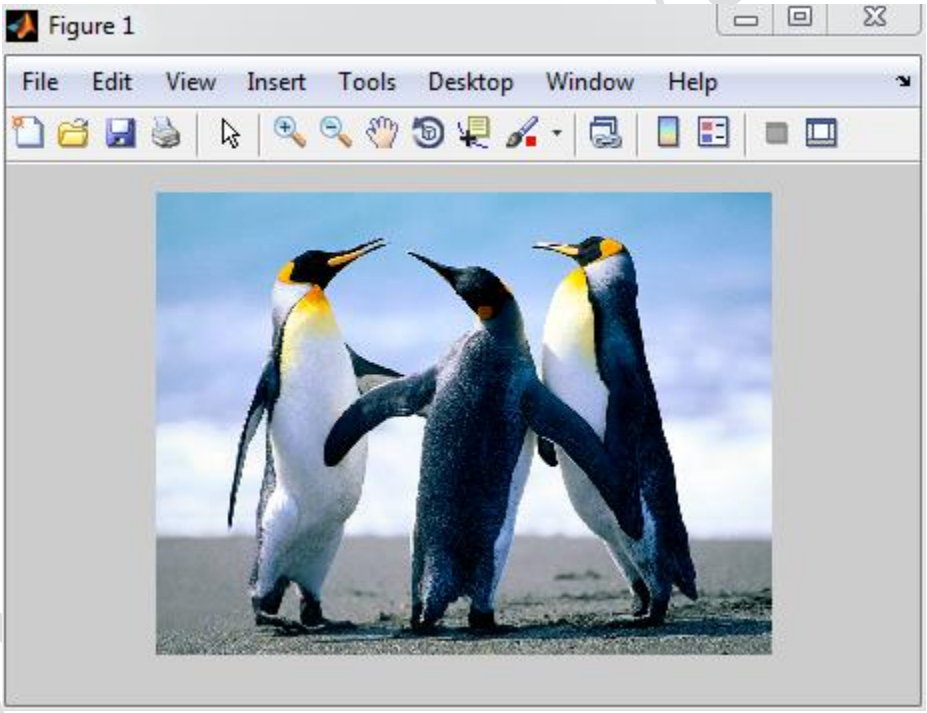
50 questions from the book Gonzalez Digital Image Processing, third edition of the chapters on the basics of digital image and intensity transformations, spatial filtering, frequency filtering, image retrieval, color image processing, and morphological image processing.

All the topics of all the chapters above in the exercises designed by me(Marjan Mavaddat) are fully covered and documented.

Example 1) Display an image and then apply salt and pepper noise to it and then use the middle filter to remove the noise?

```
img=imread('C:\Users\Marjan\Desktop\Sample
Pictures\Penguins.jpg');
figure,imshow(img);title('this is Primary picture');
```

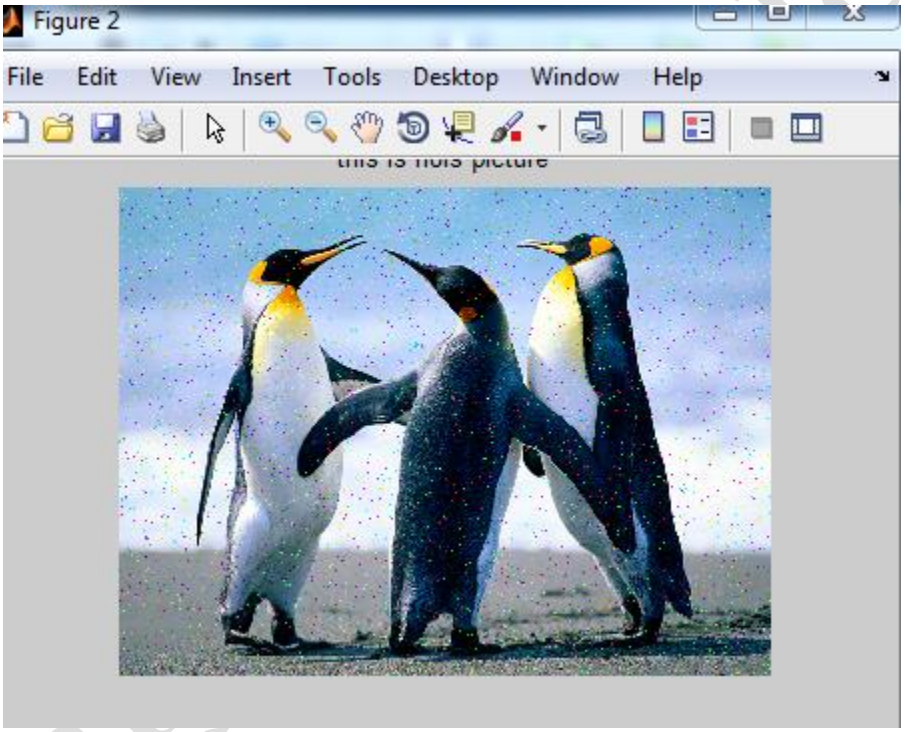


Imread reads this image function (whose address is enclosed in parentheses between quotation marks), puts it in the img variable, and displays it in a figure with the imshow function, called this is Primary picture.

Noise of salt and pepper

```
y = imnoise(img,'salt & pepper',0.02);
figure,imshow(y);title('this is nois picture');
```

Using the imnoise function and with a noise density of 0.02, salt and pepper noise is added to the image.

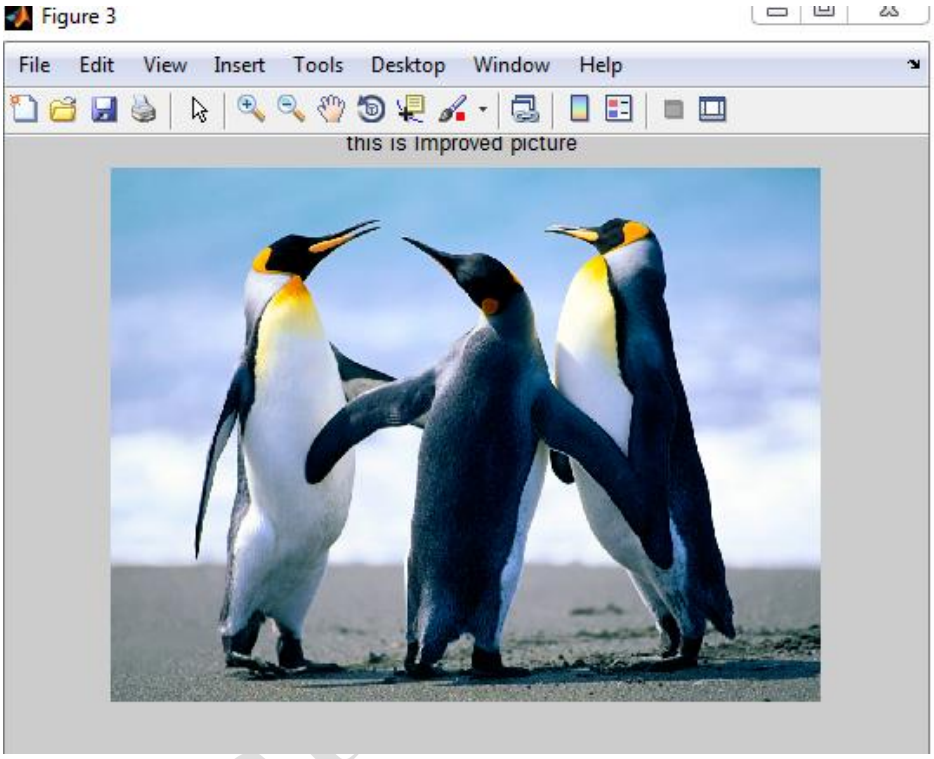


The RGB image moderate filter is applied

```
R = img(:,:,1);
G = img(:,:,2);
B = img(:,:,3);
medfilimg(:,:,1) = medfilt2(R);
medfilimg(:,:,2) = medfilt2(G);
medfilimg(:,:,3) = medfilt2(B);
```

```
figure,imshow(medfilimg);title('this is Improved
picture');
```

If the image was in color and we wanted to apply the middle filter on it, on the red, green and blue parts, the middle filter is applied separately with the medfilt2 function and a variable is stored on the color channels to get the resulting image.



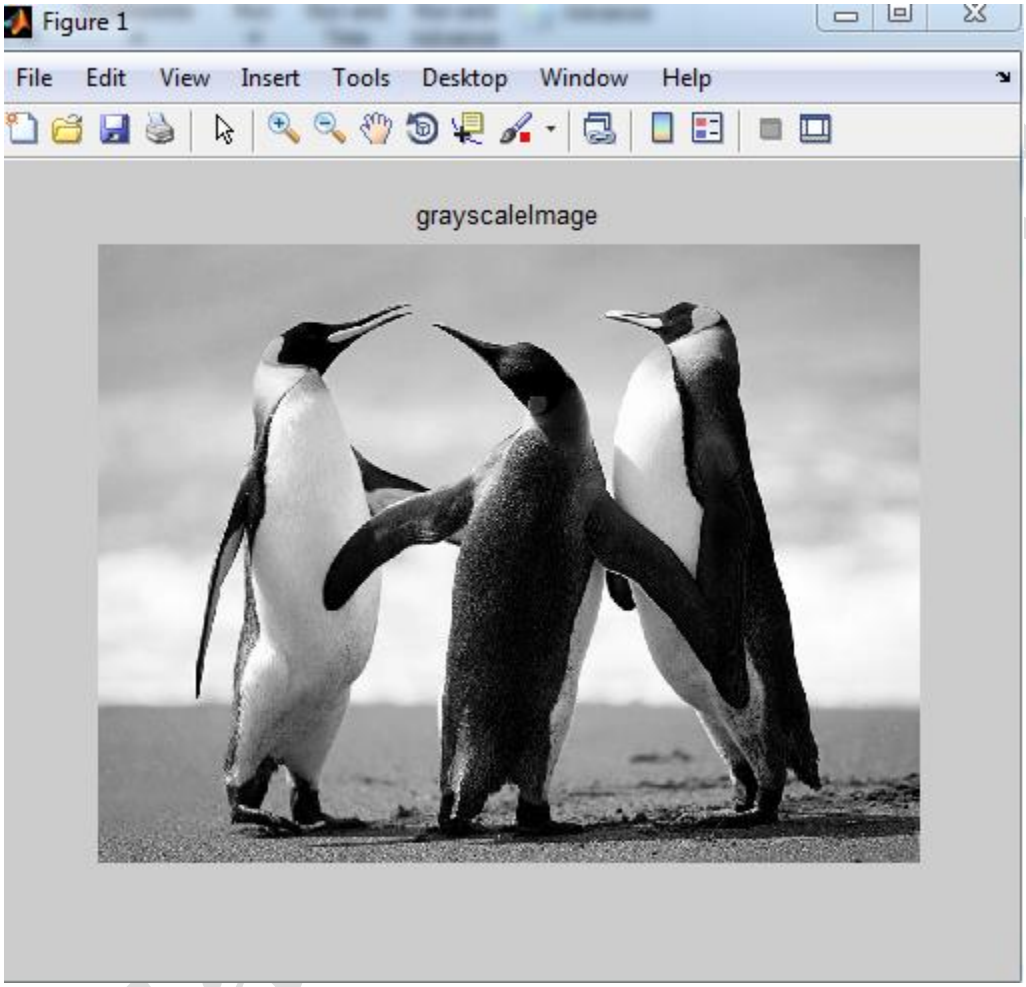
As we can see, the middle filter is able to eliminate salt and pepper (spot) noise well.

Example 2) First convert the original image to a scale image and then apply the histogram adjustment to it and display the histogram of both output images?

```
I=imread('C:\Users\Marjan\Desktop\Sample
Pictures\Penguins.jpg');
y=rgb2gray(I);
imwrite(y,'C:\Users\Marjan\Desktop\Sample
Pictures\Penguins.jpg');
I=imread('C:\Users\Marjan\Desktop\Sample
Pictures\Penguins.jpg');
imshow(I);
```

```
title('grayscaleImage')
```

The first line reads the image. The second line converts it to GrayScale with the rgb2gray function, and the next line stores it on the original image, meaning we no longer have that color image, and it turns gray. The result has the same address as the previous image. When we read and display, we see a gray image:



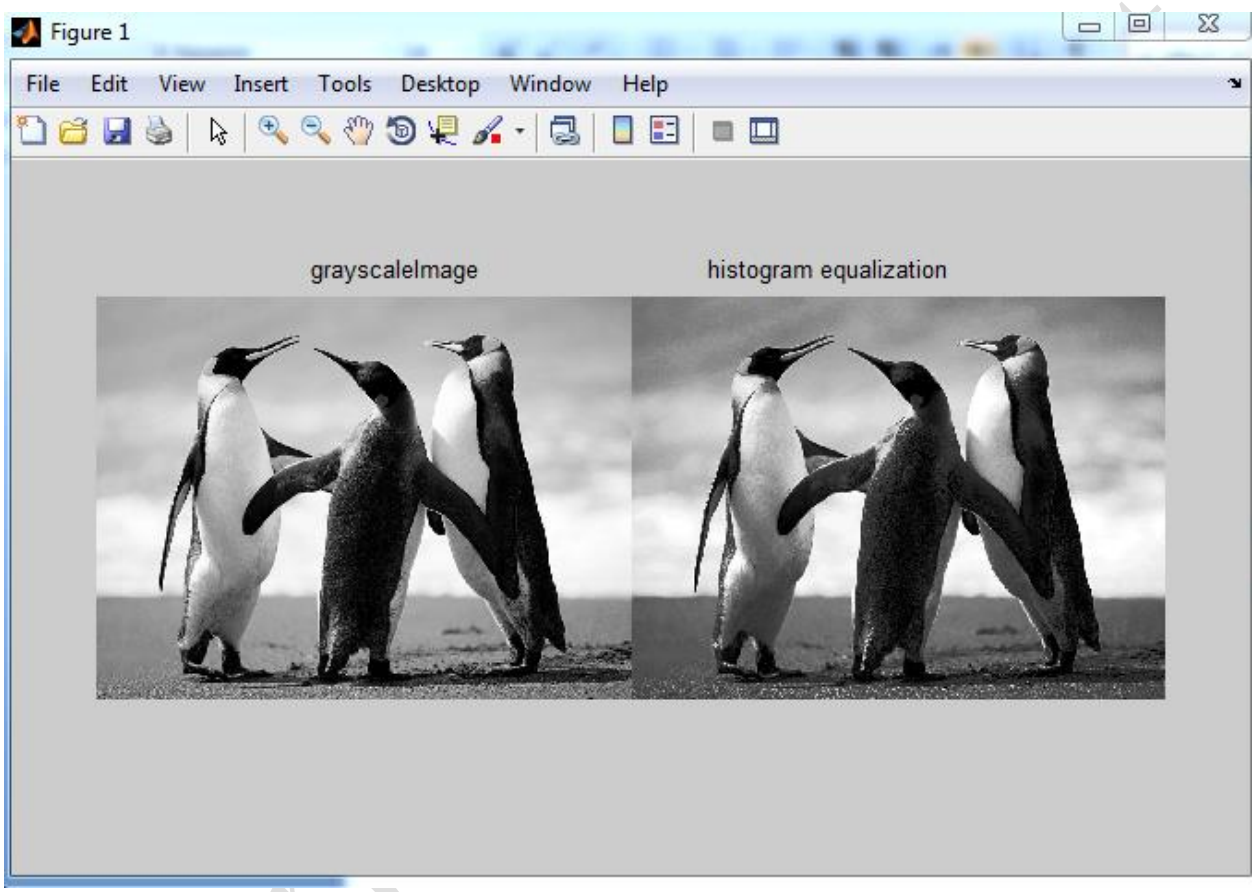
Histogram adjustment

```
J = histeq(I);
imshowpair(I,J,'montage');
title('grayscaleImage
histogram equalization');
axis off
figure
```

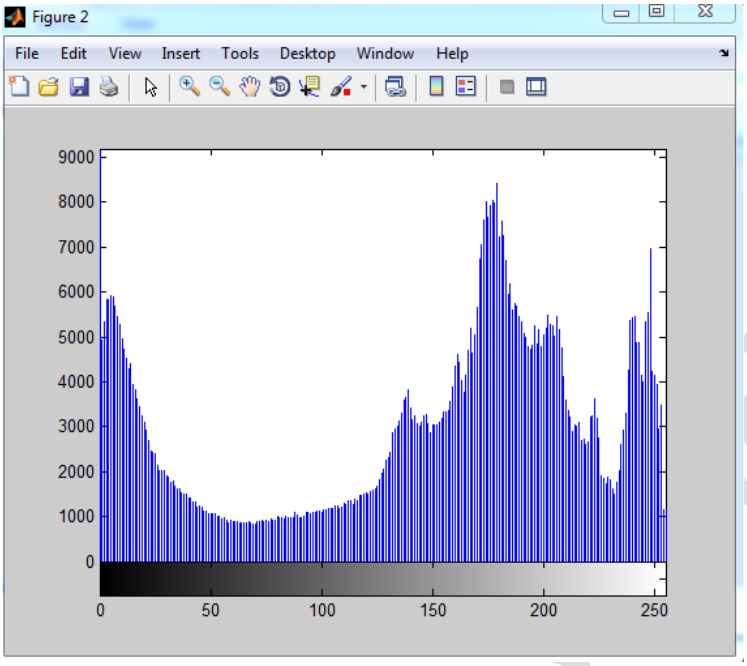
June 20, 2022

```
imhist(I,256);
figure
imhist(J,256);
```

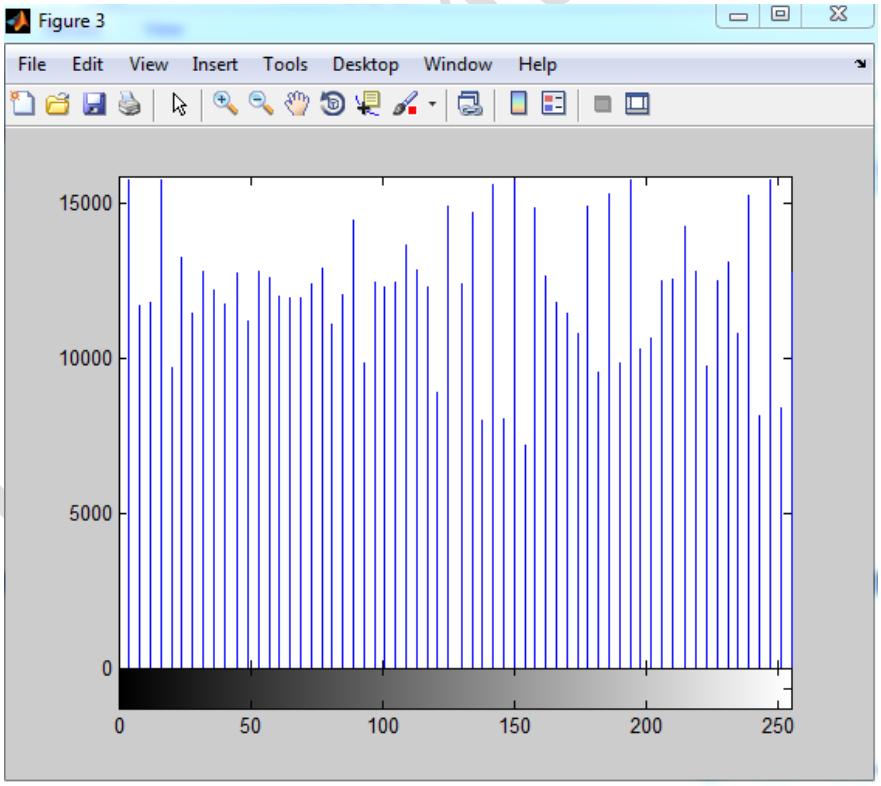
With the histeq function, we modify the image and in the next line, we display both images together, and with the imhist histogram function, we obtain both images and display them with 256 views.



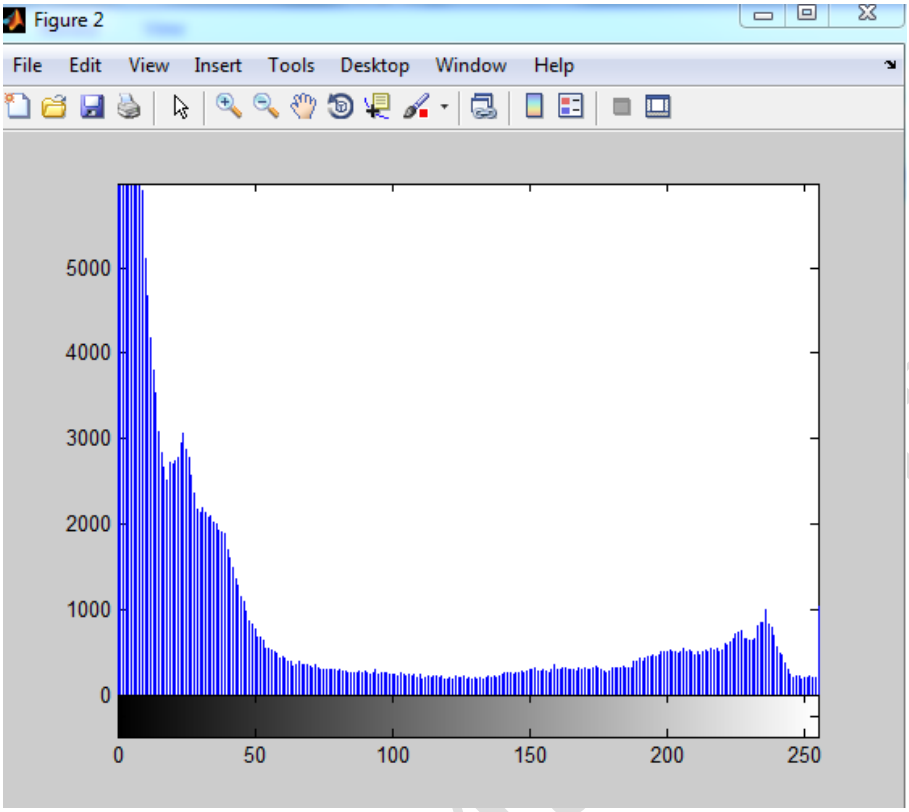
Original image histogram

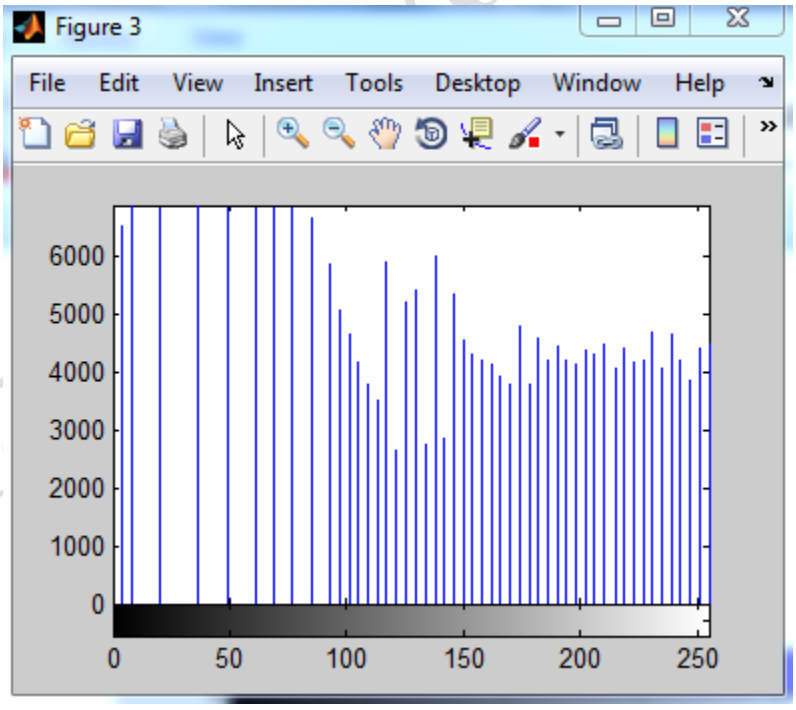


Modified image histogram



June 20, 2022





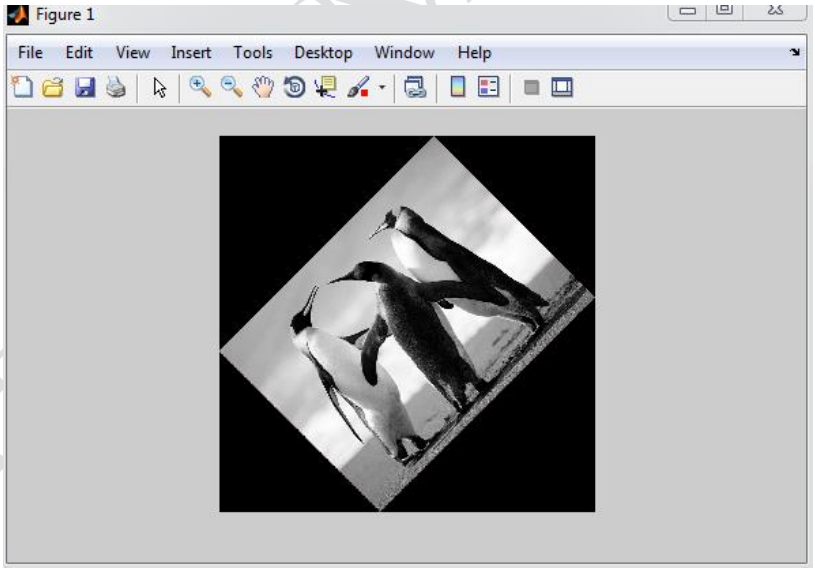
As we can see in the histogram, the adjusted image includes the intensity values with a uniform range of the image in the histogram.

June 20, 2022

Example 3) Scale illustration Rotate the previous example 45 degrees to the left and display the result?

```
A = imread('C:\Users\Marjan\Desktop\Sample Pictures\Penguins.jpg');

theta = 45;

tform = affine2d([cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1]);

outputImage = imwarp(A,tform);

figure, imshow(outputImage);
```

The first line reads the image and then we convert the 3rd conversion from page 89 of the book, which is for the period. We considered the angle to be 45 degrees. The imwarp function here performs the desired geometric transformation for us.



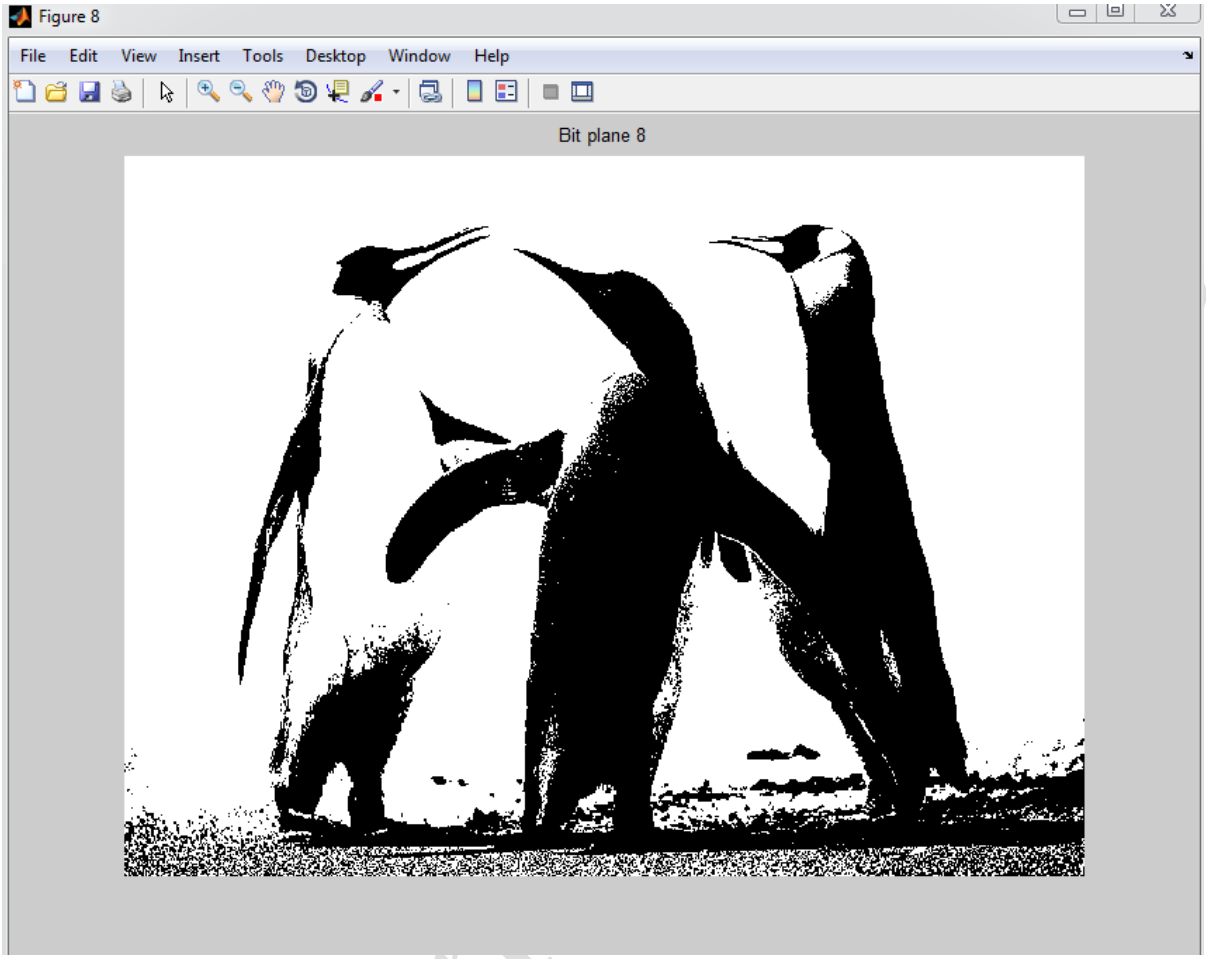
As we can see, the image is rotated 45 degrees to the left.

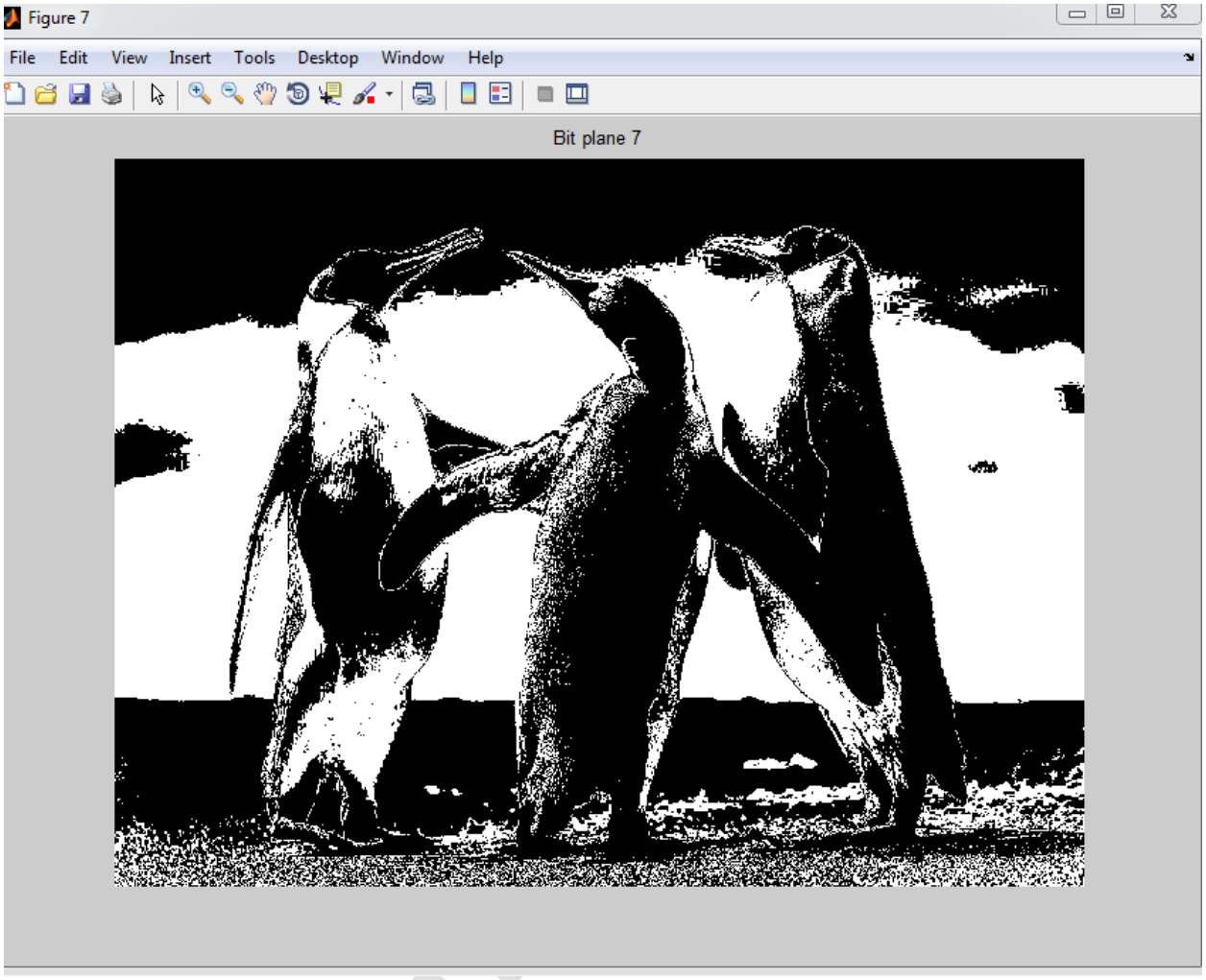
Example 4) Display the 8-bit page of Skyl Image.

```
% f=imread('C:\Users\Marjan\Desktop\Sample
Pictures\Penguins.jpg');
% figure, imshow(f);
% A=rgb2gray(f);
% figure, imshow(A);
A=imread('C:\Users\Marjan\Desktop\Sample
Pictures\Penguins.jpg');
B0=bitget(A,1); figure, imshow(logical(B0));title('Bit
plane 1');
B1=bitget(A,2); figure, imshow(logical(B1));title('Bit
plane 2');
B2=bitget(A,3); figure, imshow(logical(B2));title('Bit
plane 3');
B3=bitget(A,4); figure, imshow(logical(B3));title('Bit
plane 4');
B4=bitget(A,5); figure, imshow(logical(B4));title('Bit
plane 5');
B5=bitget(A,6); figure, imshow(logical(B5));title('Bit
plane 6');
B6=bitget(A,7); figure, imshow(logical(B6));title('Bit
plane 7');
B7=bitget(A,8); figure, imshow(logical(B7));title('Bit
plane 8');
```

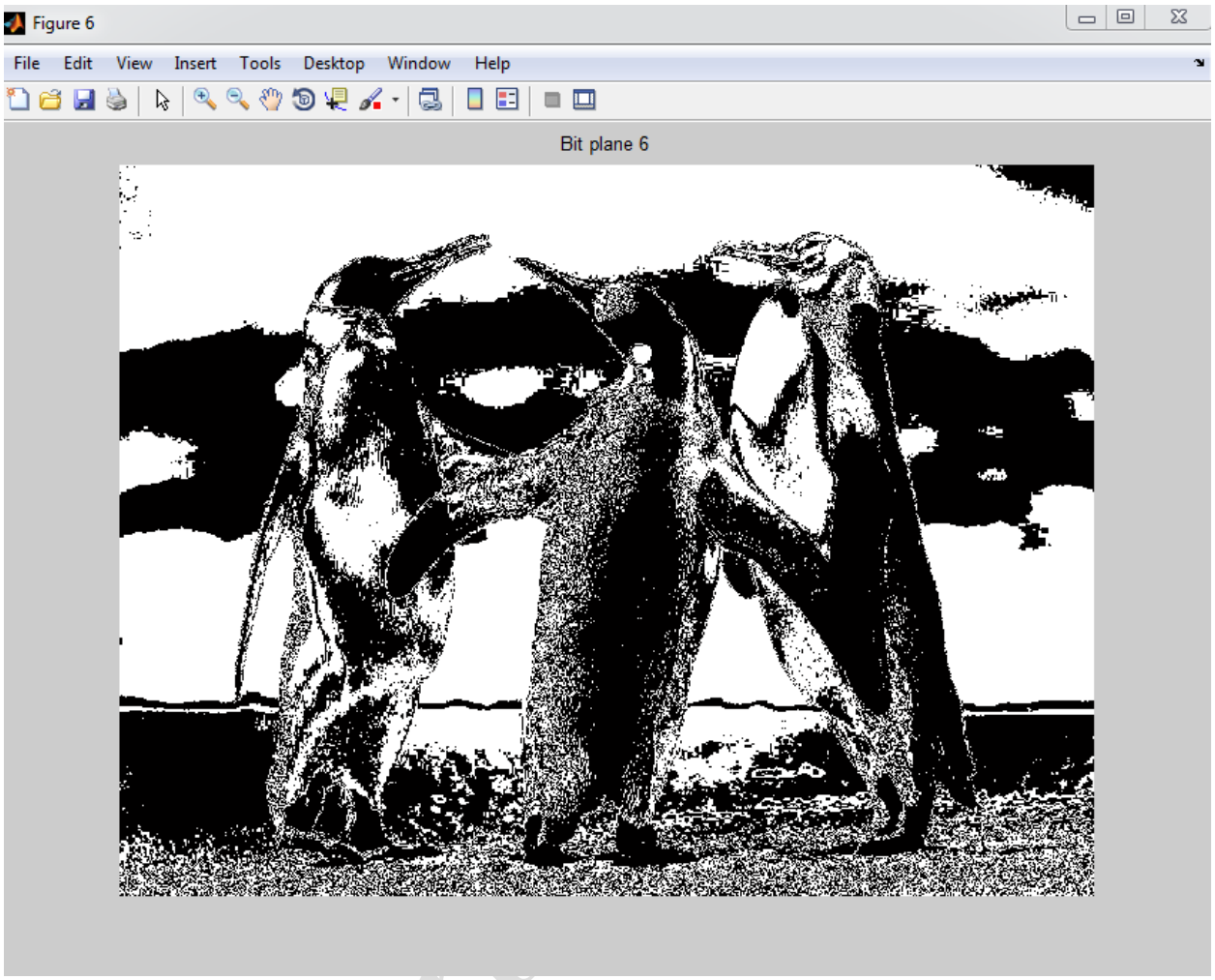
The first few lines that are commented are the image conversion to gray scale, and then with the bitget

function, we get and display bit pages 1 to 8.

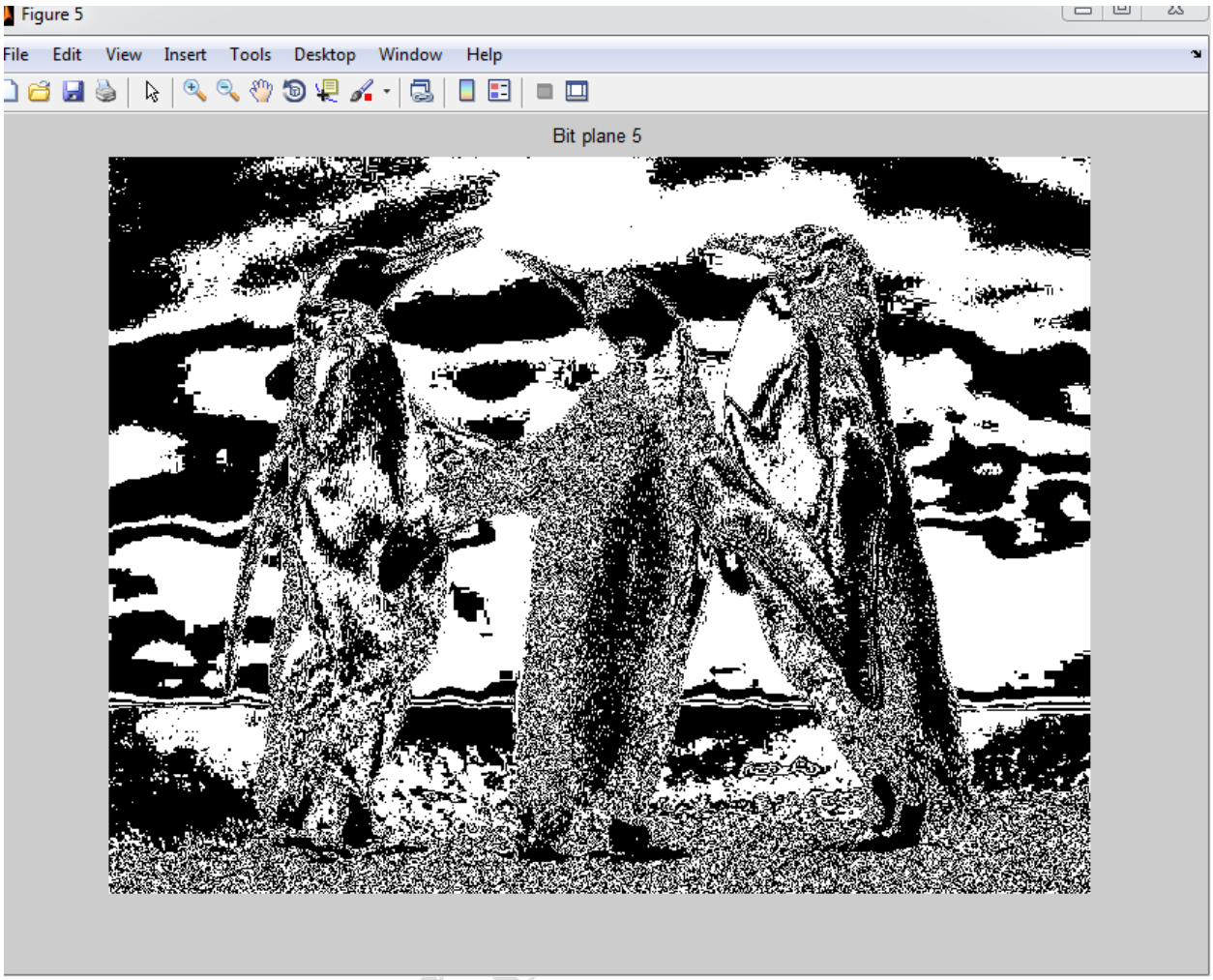




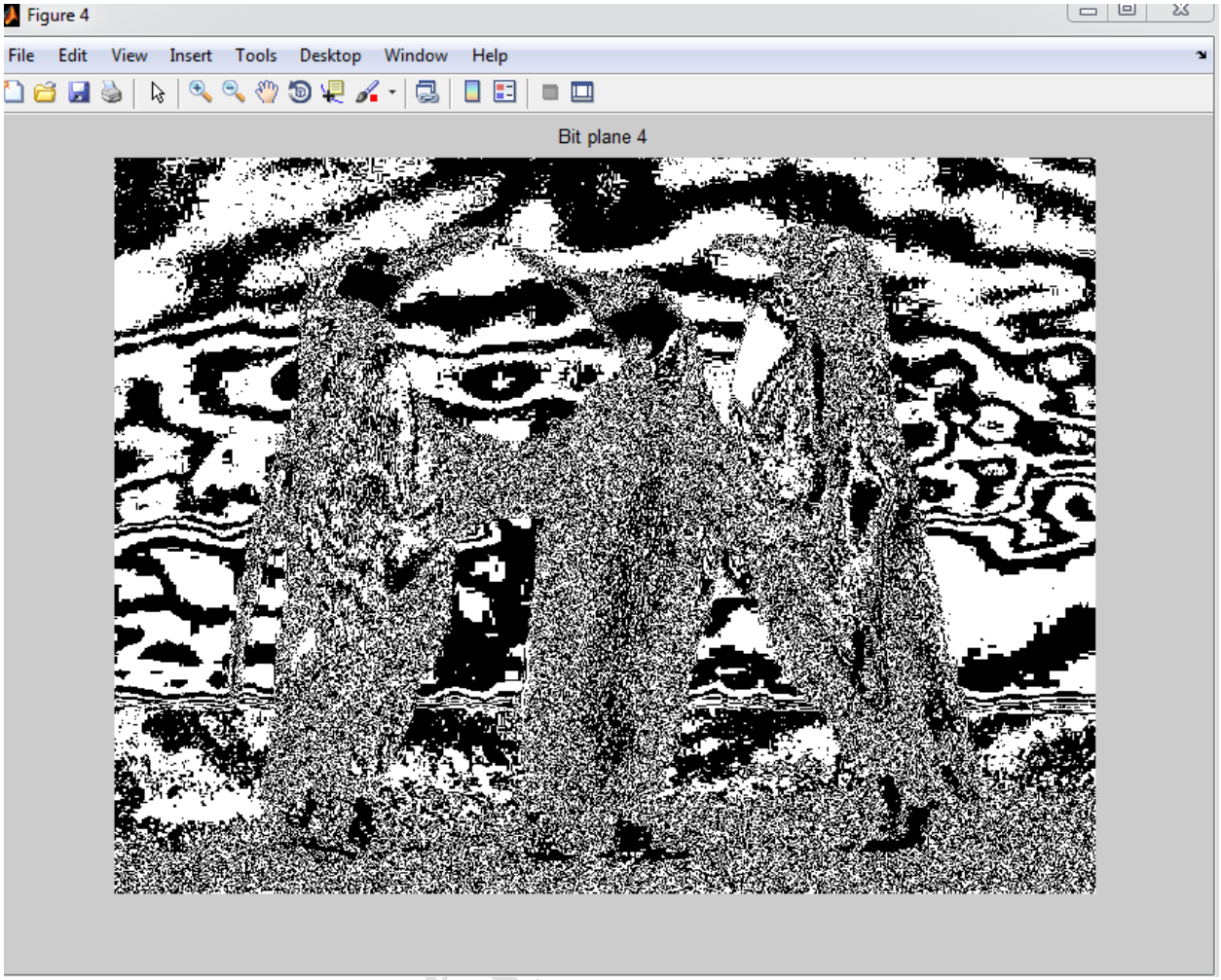
June 20, 2022



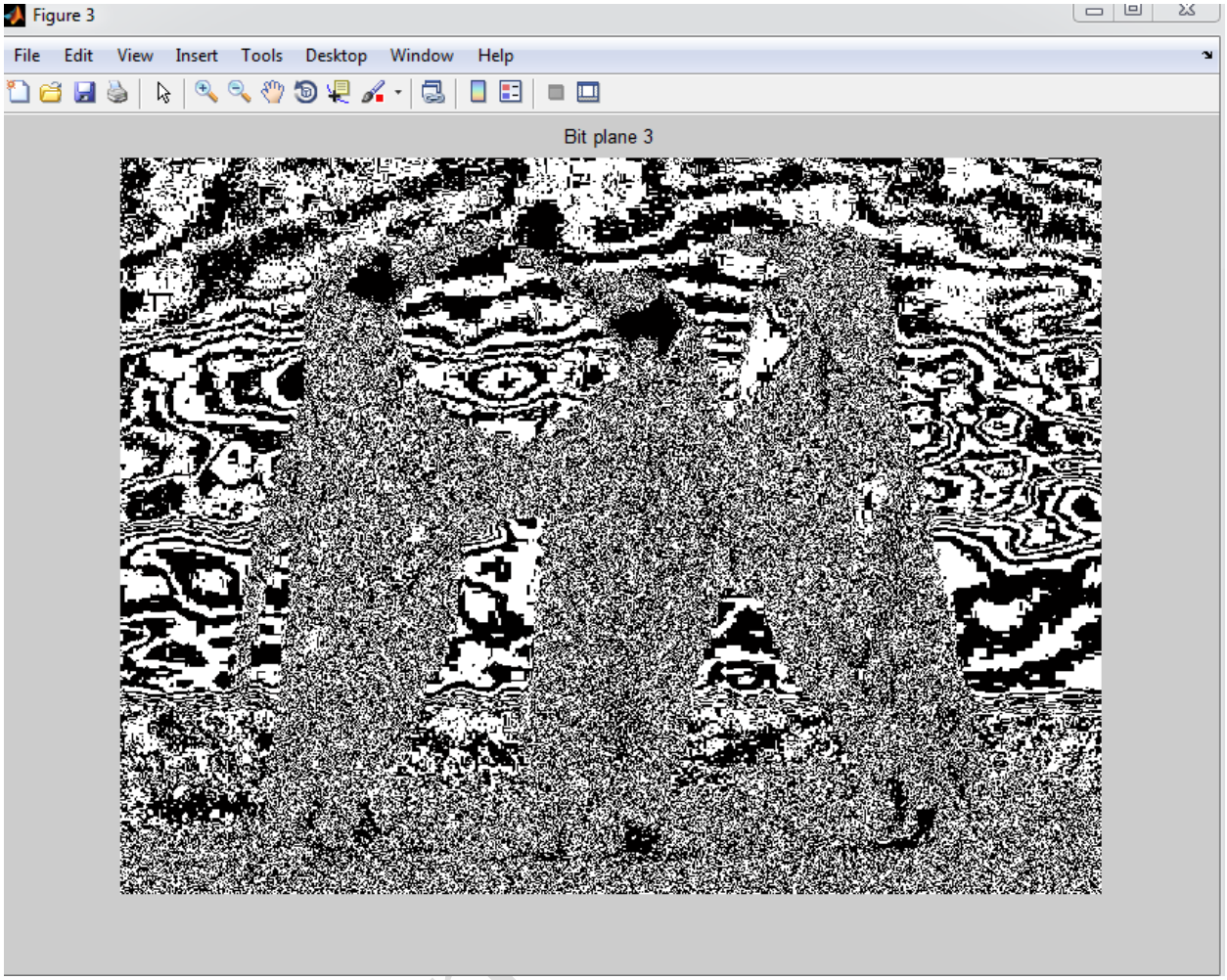
June 20, 2022



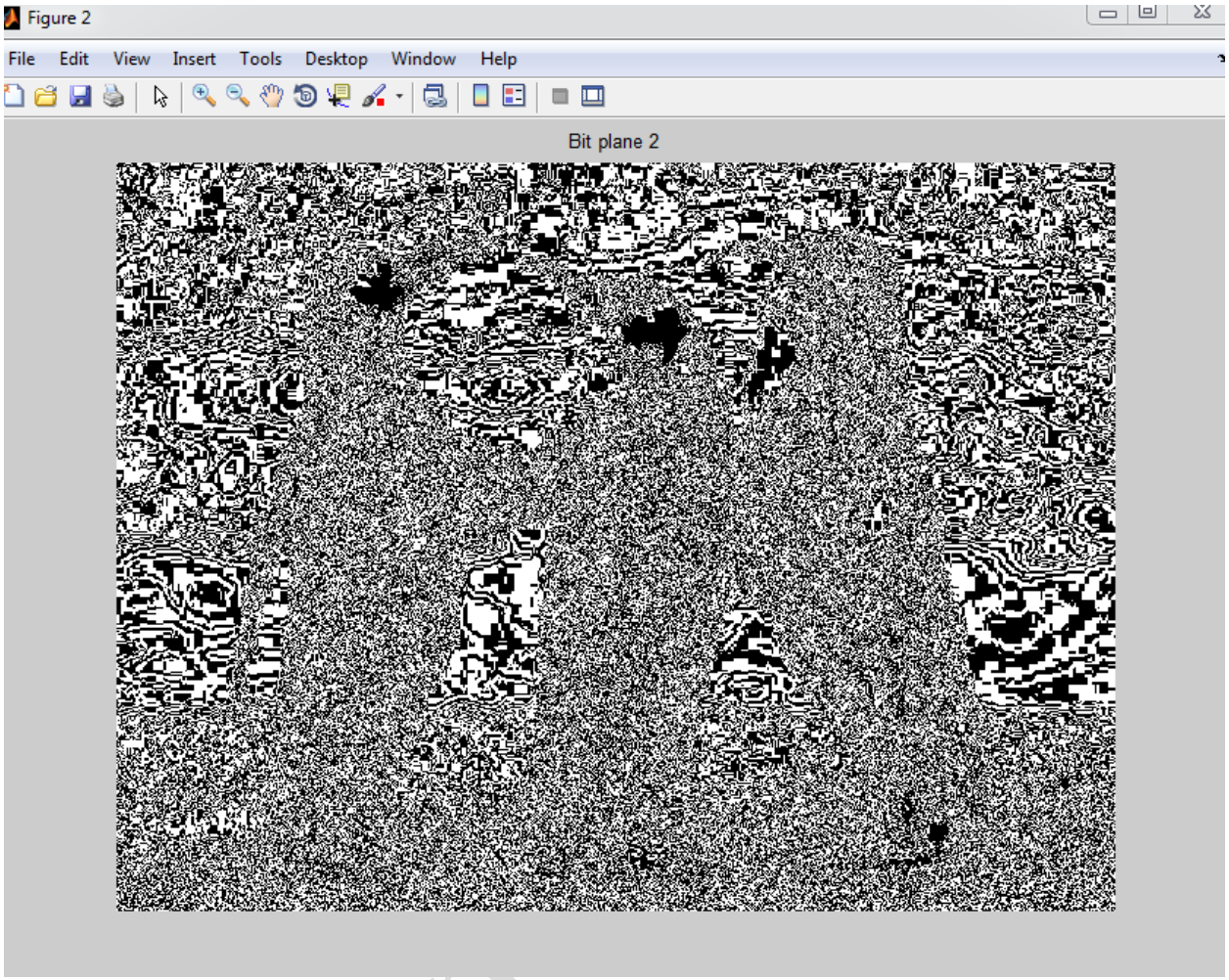
June 20, 2022



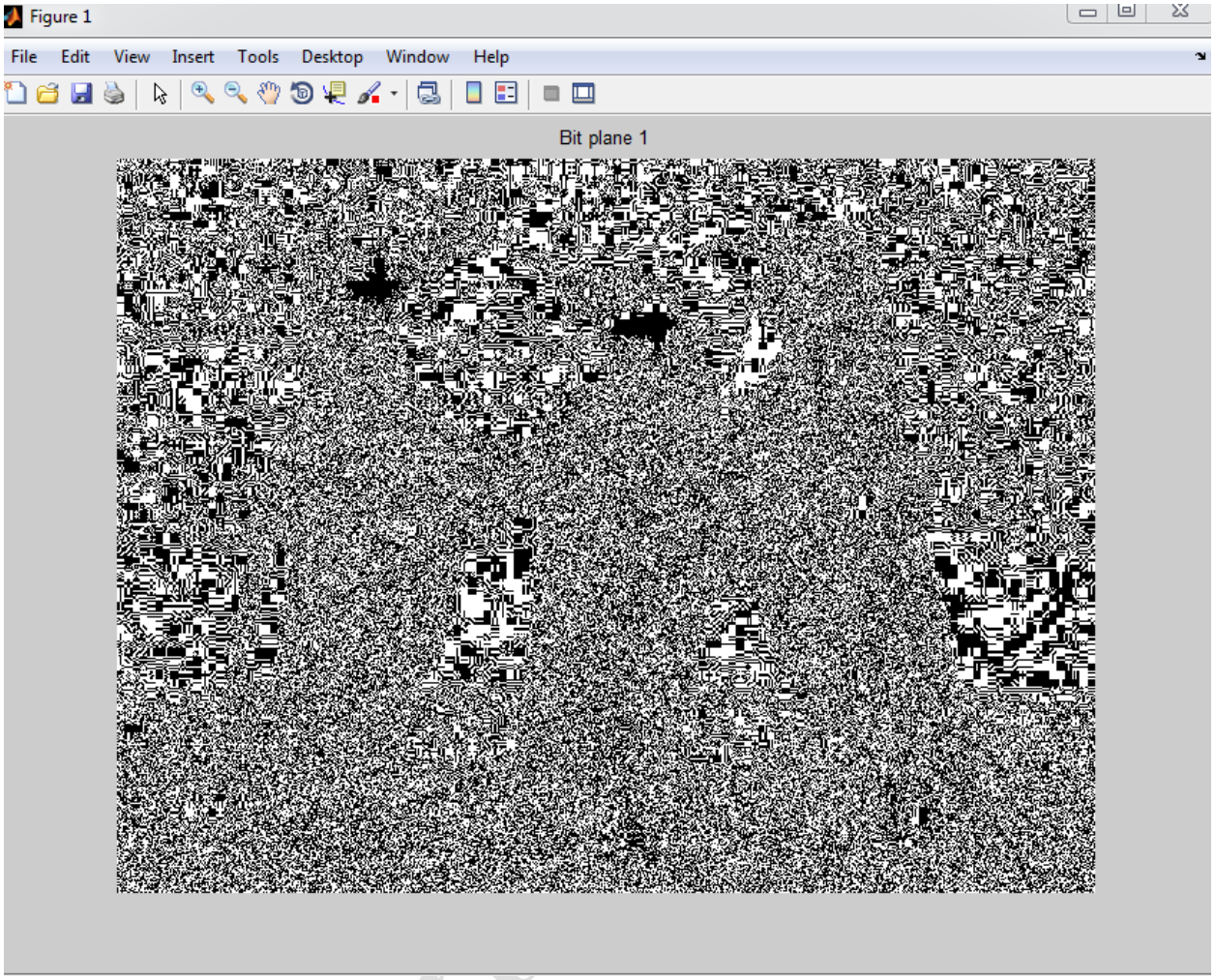
June 20, 2022



June 20, 2022



June 20, 2022



You can see Bitplan 8 to 1 in the following figures for my image, respectively:

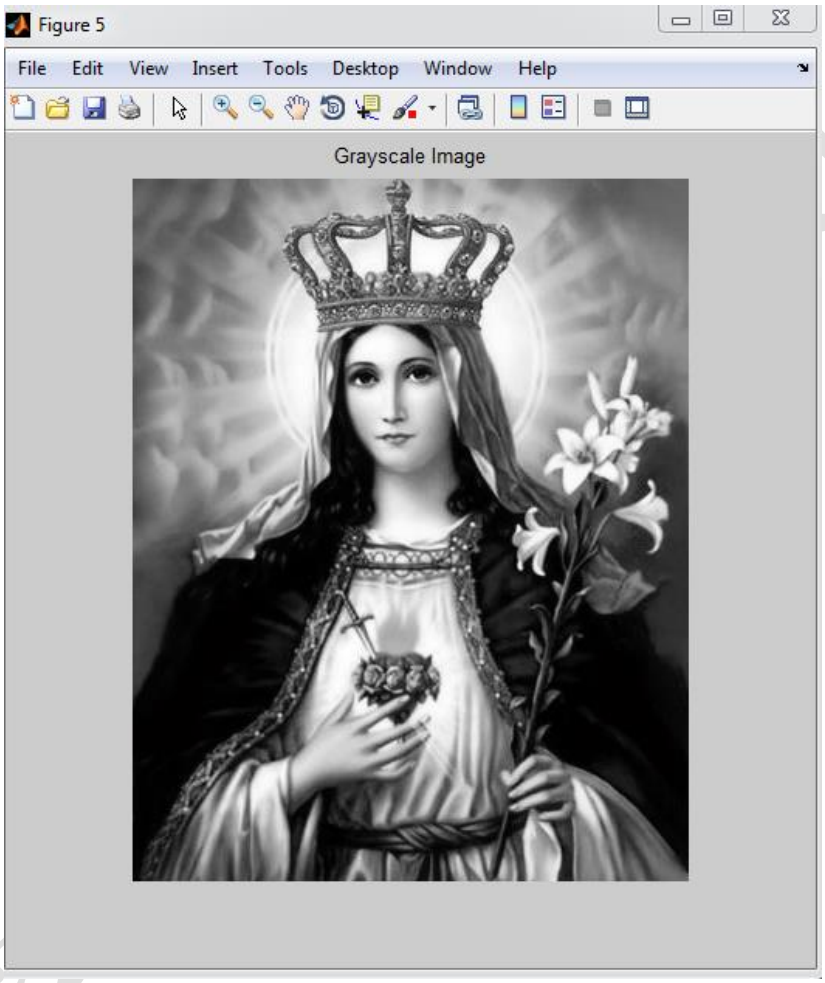
As we can see, Bitcoin 8 is most similar to the original image.

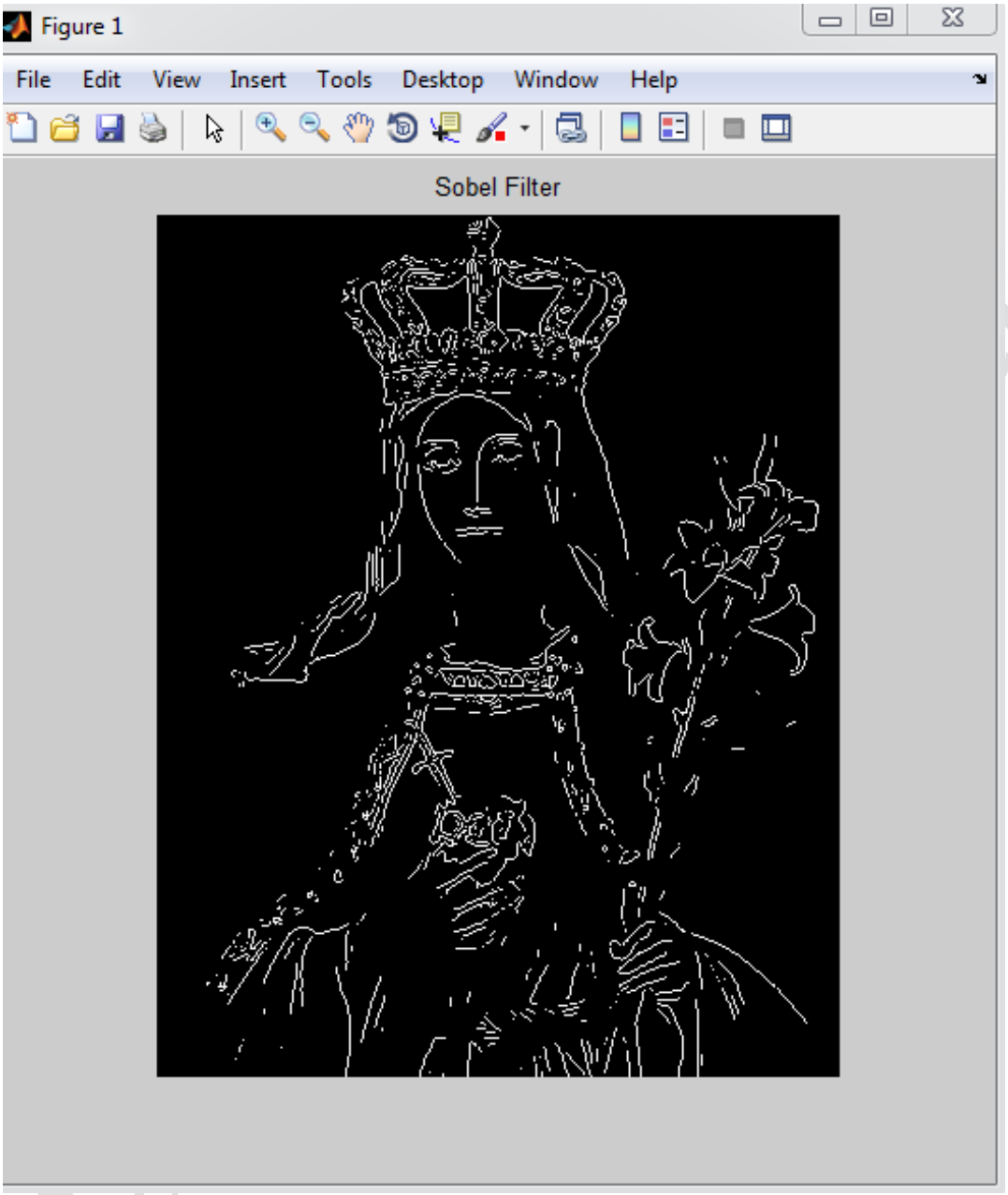
Example 5) First convert an image to gray scale and then apply Sobel, Kenny, Provit and Roberts filters to it.

```
I = imread('C:\Users\Marjan\Documents\MATLAB\q.jpg');
m=rgb2gray(I);
x = edge(m,'sobel');
y = edge(m,'canny');
z = edge(m,'prewitt');
w = edge(m,'roberts');
figure,imshow(x);title('Sobel Filter');
figure,imshow(y);title('Canny Filter');
figure,imshow(z);title('Prewitt Filter');
```

```
figure,imshow(w);title('Roberts Filter');
figure,imshow(m);title('Grayscale Image');
```

Using the edge function, we take the gray image and then apply the filter to the image with one of the filters called the second argument.





June 20, 2022

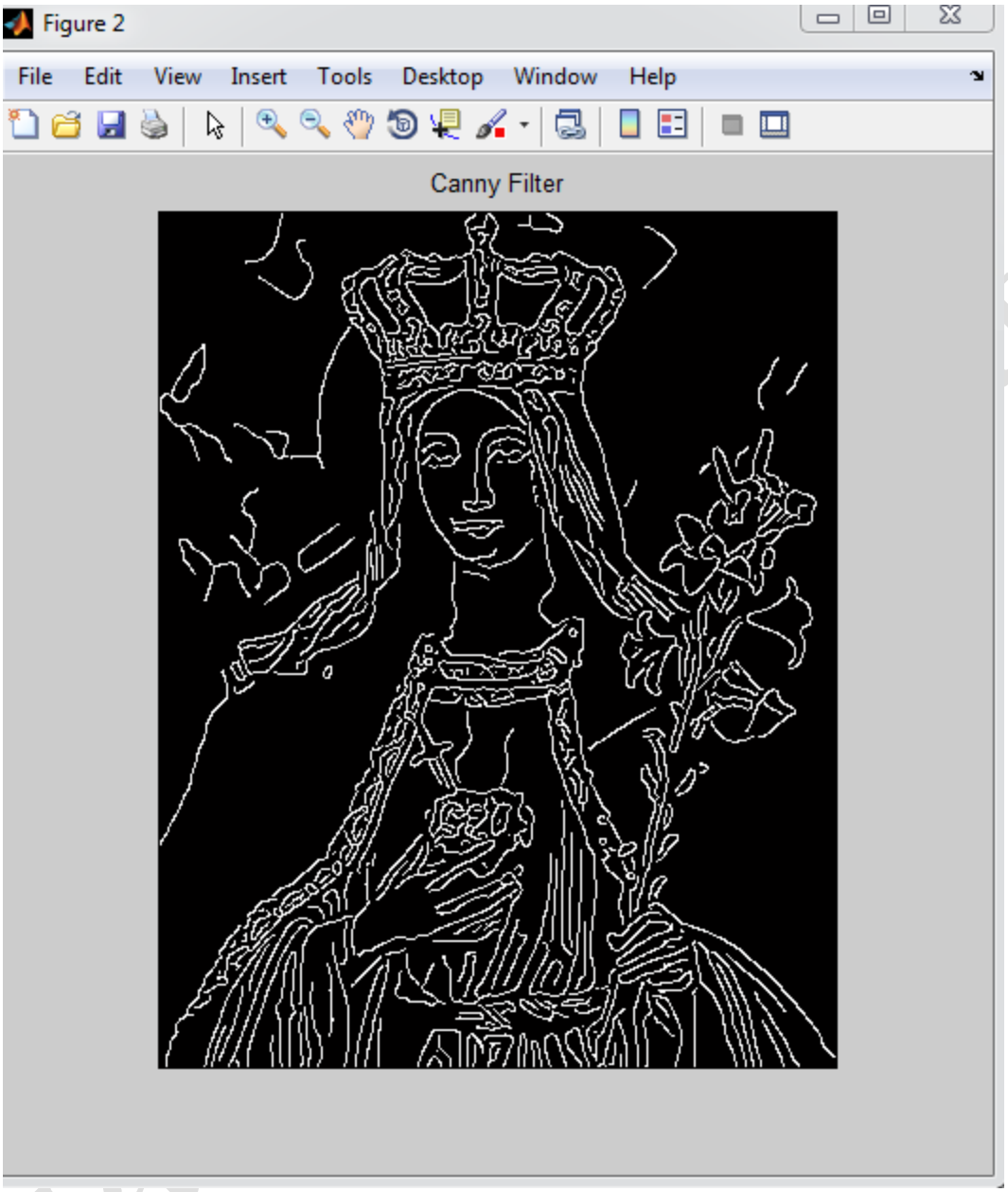


Figure 2
File Edit View Insert Tools Desktop Window Help
Canny Filter

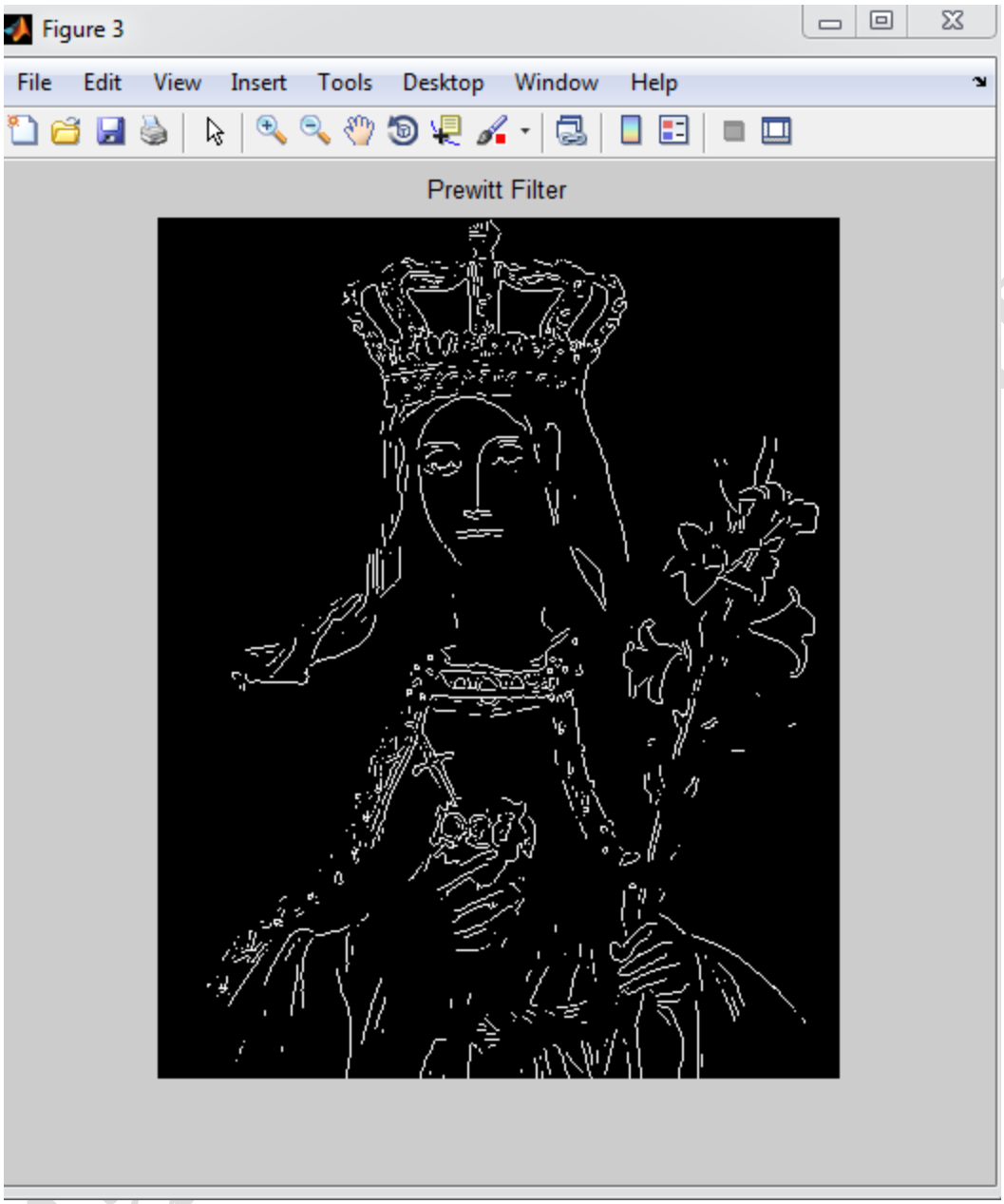
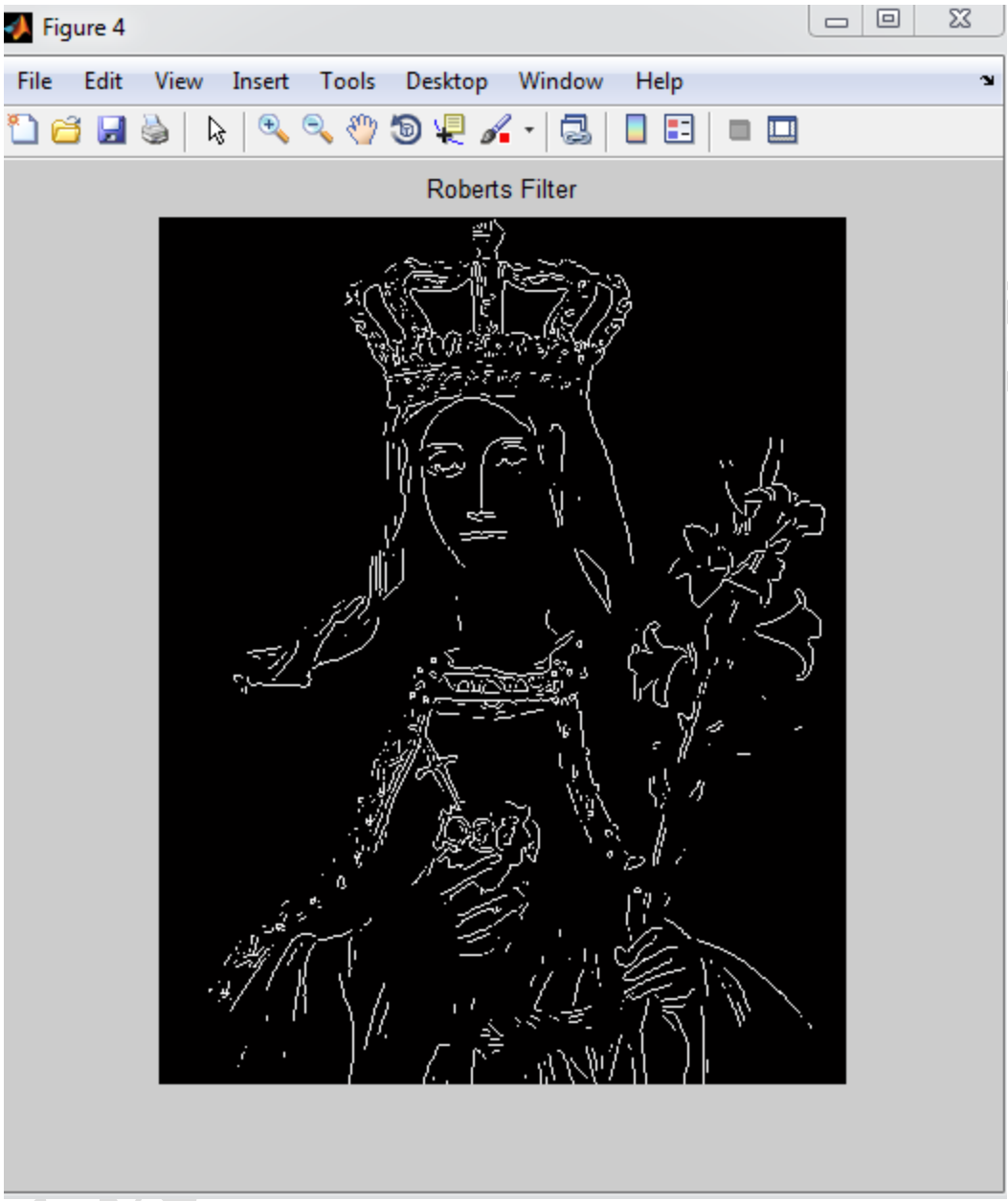


Figure 3
File Edit View Insert Tools Desktop Window Help
Prewitt Filter

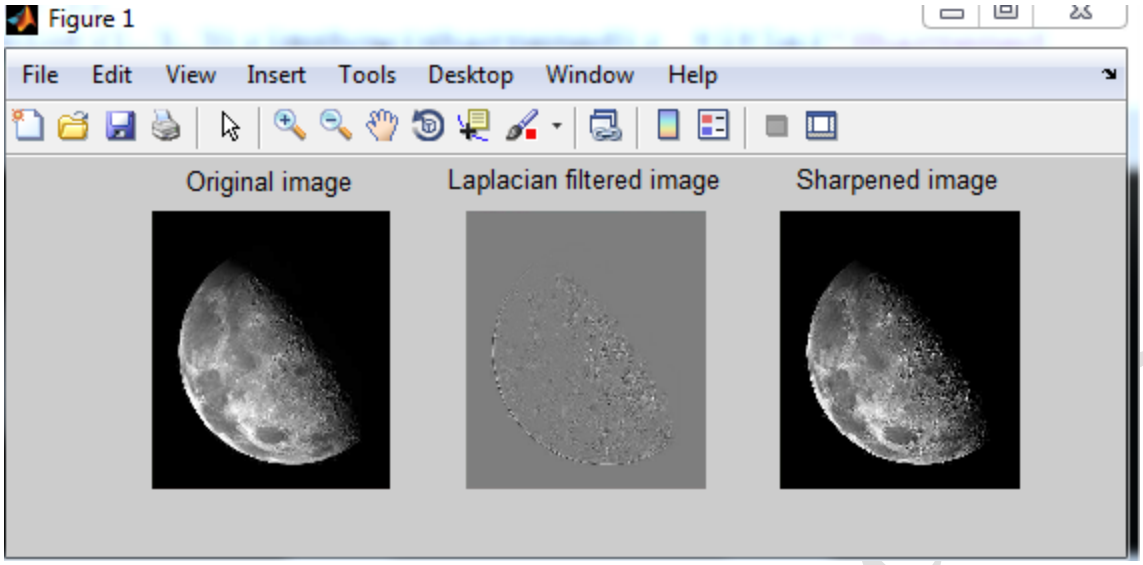


As we can see, the filter has the most edge detection compared to other filters.

Example 6) Sharpen an image using Laplace and resize.

```
clc;
close all;
a =
im2double(imread('C:\Users\Marjan\Documents\MATLAB\c.png')); %تصویر رامی خواند
```

```
lap = [-1 -1 -1; -1 8 -1; -1 -1 -1]; فیلتر لاپلاسین
resp = imfilter(a, lap, 'conv');
%اعمال فیلتر روی تصویر با کانولوشن

%نرمالسازی جواب
minR = min(resp(:));
maxR = max(resp(:));
resp = (resp - minR) / (maxR - minR);

%افزودن لبه های لاپلاسین به تصویر اصلی
sharpened = a + resp;

%نرمالسازی تصویر شارپ شده
minA = min(sharpened(:));
maxA = max(sharpened(:));
sharpened = (sharpened - minA) / (maxA - minA);

%افزایش کنتراست خطی
sharpened = imadjust(sharpened, [60/255 200/255], [0
1]);

figure; نمایش نتیجه تصویر اصلی و لاپلاسین و تصویر شارپ
شده
subplot(1,3,1);imshow(a); title('Original image');
subplot(1,3,2);imshow(resp); title('Laplacian filtered
image');
subplot(1,3,3);imshow(sharpened); title('Sharpened
image');
```



The first image is scaled from the left of the main image and the next Laplacein image, and the next image is resized plus the edges of Laplacein, which makes the image from the first image sharper.

Example 7) Fuzzy adjustment for low contrast images.

Main (reads the image and gives it to the function. Fuzzy adjustment is applied to it and the result is returned.)

```
% rgbInputImage = imread('peppers.png');
rgbInputImage =
imread('C:\Users\Marjan\Documents\MATLAB\49\50\51\1.jpg
');
% rgbInputImage =
imread('C:\Users\Marjan\Documents\MATLAB\49\50\51\Marja
n.jpg');
labInputImage =
applycform(rgbInputImage,makecform('srgb2lab'));
Lbpdfhe = fcnBPDFHE(labInputImage(:,:,1));
labOutputImage =
cat(3,Lbpdfhe,labInputImage(:,:,2),labInputImage(:,:,3)
);
rgbOutputImage =
applycform(labOutputImage,makecform('lab2srgb'));
figure, subplot 121, imshow(rgbInputImage);title('Low-
contrast image');
```

```
subplot 122,
imshow(rgbOutputImage);title('Fuzzy Equalization');
```

fcnBPDFHE.m

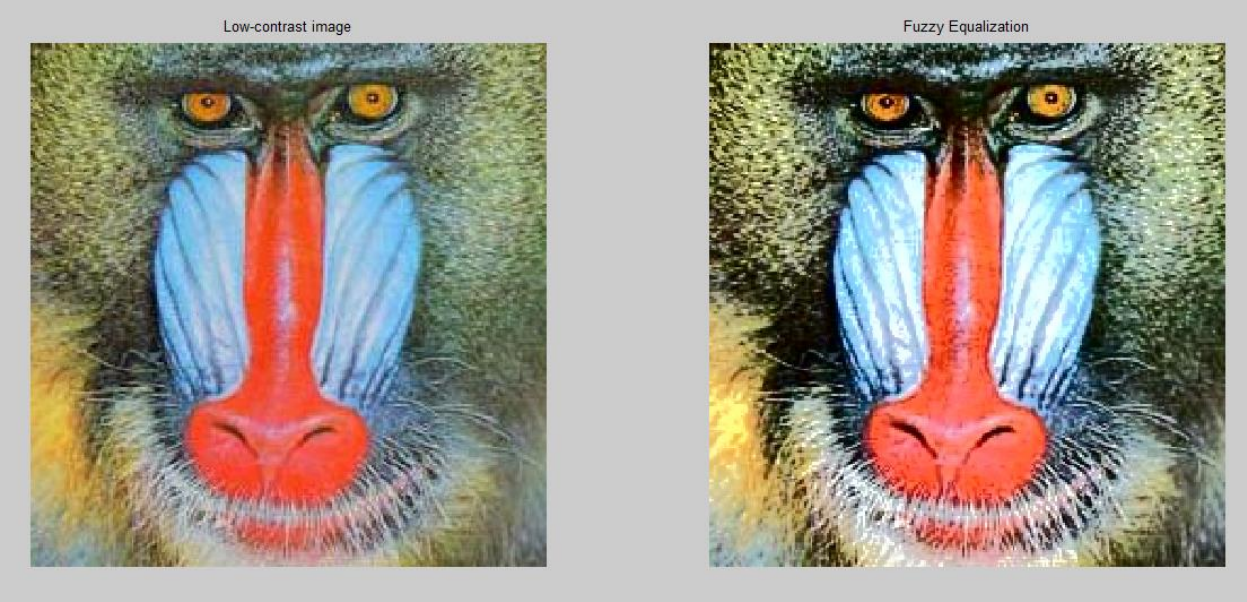
```
function [outputImage, transformationMap] =
fcnBPDFHE(inputImage, fuzzyMembershipType, parameters)
                                %بررسی پشتیبانی آرگومان ورودی
iptcheckinput(inputImage,{'uint8','uint16','int16','sin
gle','double'}, {'nonsparse','2d'}, mfilename,'I',1);
if nargin == 1
    parameters = 5;
    membership = parameters(1)-abs(-
parameters(1):parameters(1));
elseif nargin == 3
    if strcmp(fuzzyMembershipType,'triangular')
        if ~(numel(parameters)==1)
            error('fcnBPDFHE supports only 1 parameter
for Triangular Membership');
        end
        membership = parameters(1)-abs(-
parameters(1):parameters(1));
    elseif strcmp(fuzzyMembershipType,'gaussian')
        if ~(numel(parameters)==2)
            error('fcnBPDFHE supports only 2 parameters
for Gaussian Membership');
        end
        membership = exp(-(-
parameters(1):parameters(1)).^2/parameters(2)^2);
    elseif strcmp(fuzzyMembershipType,'custom')
        if numel(parameters)==0
            error('fcnBPDFHE requires the membership
value specification as 1-D array for Custom
Membership');
        end
        membership = parameters;
    else
        error('Unsupported membership type
declaration');
    end
else
```

```
    error('Unsupported calling of fcnBPDFHE');
end
imageType = class(inputImage);
%ایجاد هیستوگرام با توجه به نوع تصویر ورودی
if strcmp(class(inputImage),'uint8')
    [crispHistogram,grayScales] = imhist(inputImage);
elseif strcmp(class(inputImage),'uint16')
    crispHistogram = zeros([2^16 1]);
    for counter = 1:numel(inputImage)
        crispHistogram(inputImage(counter)+1) =
crispHistogram(inputImage(counter)+1) + 1;
    end
    grayScales = 0:(2^16 - 1);
elseif strcmp(class(inputImage),'int16')
    crispHistogram = zeros([2^16 1]);
    for counter = 1:numel(inputImage)
        crispHistogram(inputImage(counter)+32769) =
crispHistogram(inputImage(counter)+32769) + 1;
    end
    grayScales = -32768:32767;
elseif
(strcmp(class(inputImage),'double')||strcmp(class(input
Image),'single'))
    maxGray = max(inputImage(:));
    minGray = min(inputImage(:));
    inputImage = im2uint8(mat2gray(inputImage));
    [crispHistogram,grayScales] = imhist(inputImage);
end
inputImage = double(inputImage);
fuzzyHistogram =
zeros(numel(crispHistogram)+numel(membership)-1,1);
for counter = 1:numel(membership)
    fuzzyHistogram = fuzzyHistogram +
membership(counter)*[zeros(counter-
1,1);crispHistogram;zeros(numel(membership)-
counter,1)];
end
fuzzyHistogram =
fuzzyHistogram(ceil(numel(membership)/2):end-
floor(numel(membership)/2));
```

```
del1FuzzyHistogram = [0;(fuzzyHistogram(3:end)-
fuzzyHistogram(1:end-2))/2;0];
del2FuzzyHistogram = [0;(del1FuzzyHistogram(3:end)-
del1FuzzyHistogram(1:end-2))/2;0];
locationIndex = (2:numel(fuzzyHistogram)-1)'+1;
maxLocAmbiguous =
locationIndex(((del1FuzzyHistogram(1:end-
2).*del1FuzzyHistogram(3:end))<0) &
(del2FuzzyHistogram(2:end-1)<0));
counter = 1;
maxLoc = 1;
while counter < numel(maxLocAmbiguous)
    if
(maxLocAmbiguous(counter)==(maxLocAmbiguous(counter+1)-
1))
        maxLoc = [maxLoc ;
(maxLocAmbiguous(counter)*(fuzzyHistogram(maxLocAmbiguo
us(counter))>fuzzyHistogram(maxLocAmbiguous(counter+1))
)) +
(maxLocAmbiguous(counter+1)*(fuzzyHistogram(maxLocAmbig
uous(counter))<=fuzzyHistogram(maxLocAmbiguous(counter+
1))))];
        counter = counter + 2;
    else
        maxLoc = [maxLoc ; maxLocAmbiguous(counter)];
        counter = counter + 1;
    end
end
if(maxLoc(end)~=numel(fuzzyHistogram))
    maxLoc = [maxLoc ; numel(fuzzyHistogram)];
end
low = maxLoc(1:end-1);
high = [maxLoc(2:end-1)-1;maxLoc(end)];
span = high-low;
cumulativeHistogram = cumsum(fuzzyHistogram);
M = cumulativeHistogram(high)-cumulativeHistogram(low);
factor = span .* log10(M);
range = max(grayScales)*factor/sum(factor);
transformationMap = zeros(numel(grayScales),1);
for counter = 1:length(low)
```

```
    for index = low(counter):high(counter)
        transformationMap(index) = round((low(counter)-
1) +
(range(counter)*(sum(fuzzyHistogram(low(counter):index)
))/(sum(fuzzyHistogram(low(counter):high(counter))))));
    end
end
outputImage = transformationMap(inputImage+1);
outputImage =
mean(inputImage(:))/mean(outputImage(:))*outputImage;
outputImage = cast(outputImage,imageType);
if strcmp(imageType,'single')
    outputImage = minGray + (maxGray-
minGray)*mat2gray(outputImage);
elseif strcmp(imageType,'double')
    outputImage = minGray + (maxGray-
minGray)*mat2gray(outputImage);
end
```

خروجی:

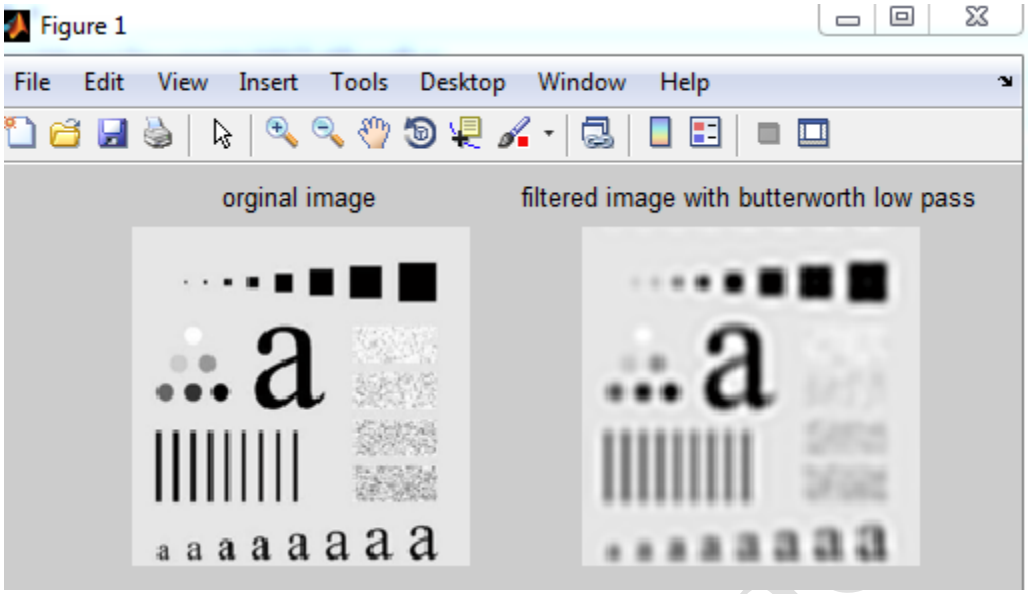


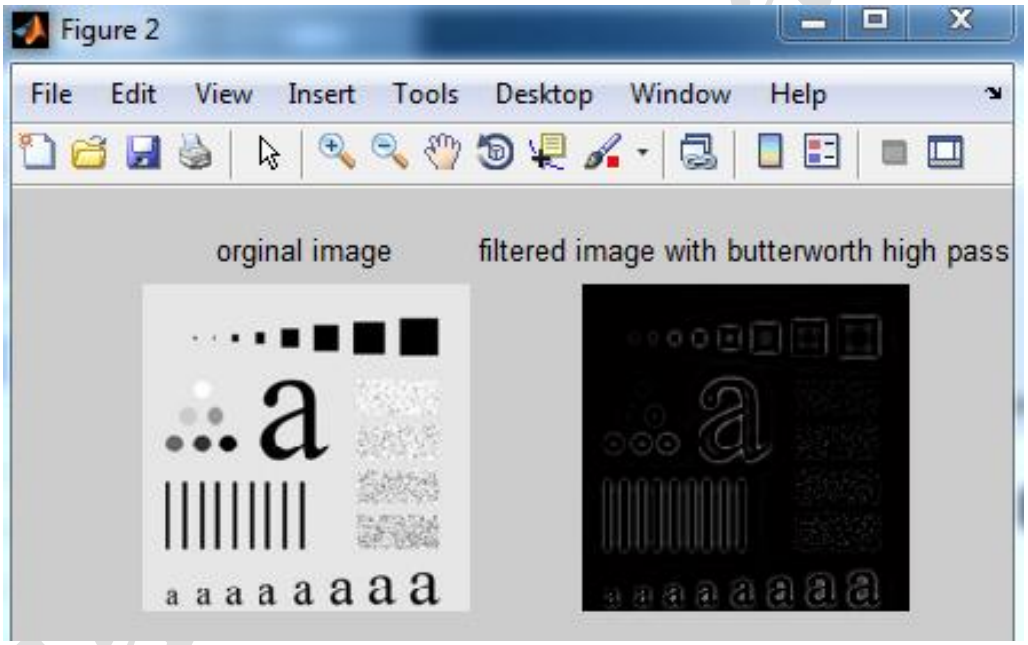
We see that fuzzy adjustment is associated with an increase in image contrast.

Question 8) Display the low-pass, high-pass, and mid-pass butterfly filter and remove the band from the 2nd order and the cut-off frequency of 400.

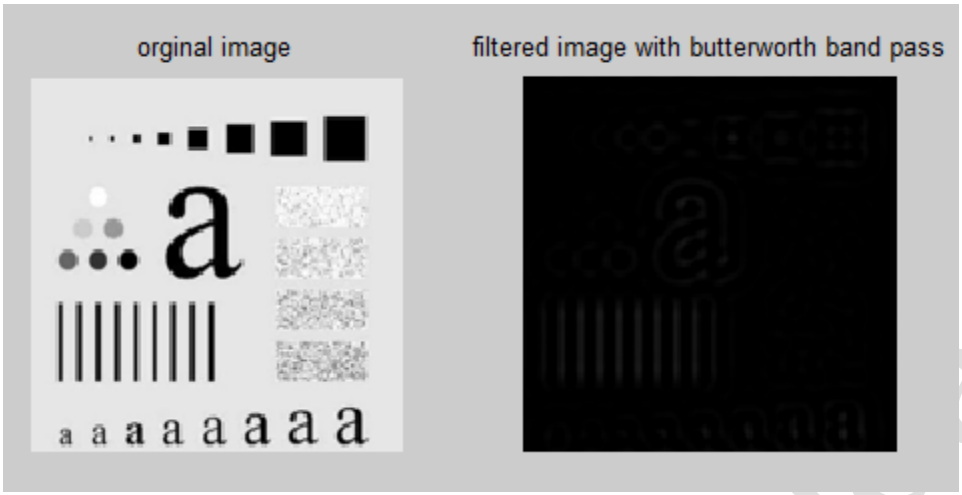
```
clc;
clear;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I=imread('C:\Users\Marjan\Documents\MATLAB\Marjan.jpg')
;
I=rgb2gray(I);
I=imresize(I,[150 150]);
[m, n]=size(I);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r1=fftshift(fft2(I));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D0=400;% فرکانس قطع
nn=2;% مرتبه
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u=0:(m-1);
v=0:(n-1);
idx=find(u>m/2);
u(idx)=u(idx)-m;
idy=find(v>n/2);
v(idx)=v(idx)-n;
[v,u]=meshgrid(v,u);
D=u.^2+v.^2;
D=fftshift(D);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
new=zeros(m,n);
for u=1:m
    for v=1:n
        new(u,v)=1/(1+(D(u,v)/D0)^(2*nn));
% فرمول فیلتر پایین گذر باتروورث
    end
end
g=uint8(ifft2(ifftshift(new.*r1)));
subplot(1,2,1);
imshow(uint8(I));
title('orginal image');
subplot(1,2,2);
imshow(uint8(g));
```

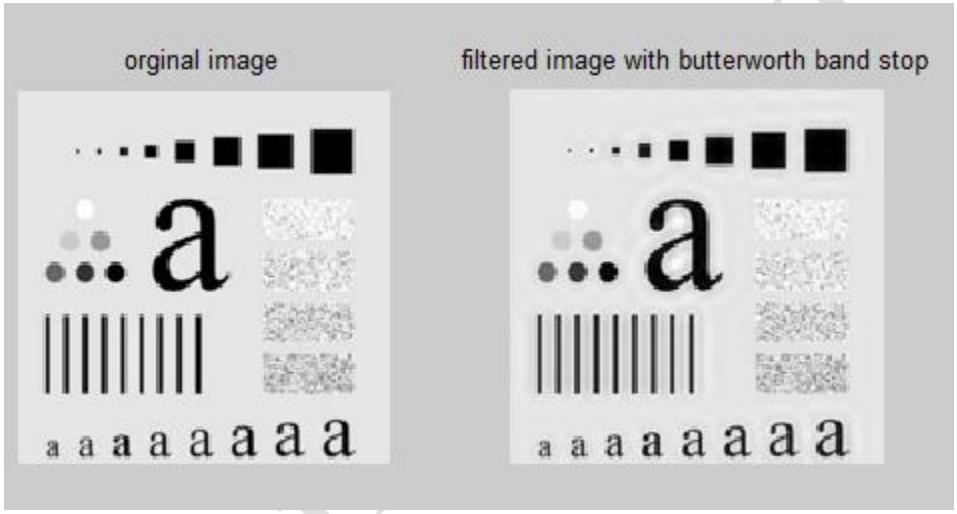
```
title('filtered image with butterworth low pass');
figure
new2=1-new;
% فرمول فیلتر بالا گذر باتروورث
g=uint8(ifft2(ifftshift(new2.*r1)));
subplot(1,2,1);
imshow(uint8(I));
title('orginal image');
subplot(1,2,2);
imshow(uint8(g));
title('filtered image with butterworth high pass');
figure
new3=new.*new2;
% فرمول فیلتر میان گذر باتروورث
g=uint8(ifft2(ifftshift(new3.*r1)));
subplot(1,2,1);
imshow(uint8(I));
title('orginal image');
subplot(1,2,2);
imshow(uint8(g));
title('filtered image with butterworth band pass');
figure
new4=1-new3;
% فرمول فیلتر حذف باند باتروورث

g=uint8(ifft2(ifftshift(new4.*r1)));
subplot(1,2,1);
imshow(uint8(I));
title('orginal image');
subplot(1,2,2);
imshow(uint8(g));
title('filtered image with butterworth band stop');
```





June 20, 2022





Take the image from the input. If the image was a color, convert it to a gray scal. Using the fft2 function, we take a two-dimensional Fourier transform image. Using the fftshift function, we place the high frequencies in the corners and the low frequencies in the center. I considered the frequency radius to be 400. In this case, we considered the order 2. I created a new matrix the same size as my Fourier transform. D The matrix is related to my distances. When my matrix is created, I multiply it in my Fourier transform image, take the Butterworth filter formula, and finally take the Fourier transform image. For the Fourier transform image, we use the ifft2 function, which I shifted.

Low pass filter passes low frequencies (blurs, like averaging)

Low pass filter = low pass-1 passes high frequencies (sharpens, edge detection)

June 20, 2022

Intermediate filter = Overpass filter * Low-pass filter (such as notch filter passes for good intermittent noise removal or passes a certain interval. It usually passes high frequencies but passes low frequencies from one place to zero. Slowly.)

Band removal filter = transient filter - 1 (removes a certain range of frequencies.)

The rotating state that is in ideal filters is reduced or eliminated in Butterworth.

Question 9) Display the low-pass and high-pass filter and the middle pass and remove the Gaussian band with a cut-off frequency of 20.

```
clc
close all
clear all
%
RGB=imread('C:\Users\Marjan\Documents\MATLAB\Marjan.jpg
');
RGB=imread('C:\Users\Marjan\Documents\MATLAB\test.jpg')
;
I=rgb2gray(RGB); تبدیل تصویر به خاکستری
A = fft2(double(I)); محاسبه تبدیل فوریه تصویر خاکستری
A1=fftshift(A); اسکیل کردن فرکانسی
محاسبه پاسخ فیلتر گوسی
[M, N]=size(A); سایز تصویر
D0=20; پارامتر سایز فیلتر(فرکانس قطع)
X=0:N-1;
Y=0:M-1;
[X, Y]=meshgrid(X,Y);
Cx=0.5*N;
Cy=0.5*M;
Lo=exp(-((X-Cx).^2+(Y-Cy).^2)./(2*D0).^2); فیلتر پایین
گذر گوسی
Hi=1-Lo; فیلتر بالاگذر filter=1-low pass filter
% Filtered image=ifft(filter response*fft(original
image))
J=A1.*Lo;ضرب تصویر در فیلتر
J1=ifftshift(J); اسکیل کردن
B1=ifft2(J1);عکس تبدیل فوریه
K=A1.*Hi; برای بالاگذر به همان صورت
```

```
K1=ifftshift(K);
B2=ifft2(K1);
K2=Lo.*Hi;برای میانگذر به همان صورت
J2=A1.*K2;
K3=ifftshift(J2);
B3=ifft2(K3);
K4=1-K2;برای حذف باند به همان صورت
J3=A1.*K4;
K4=ifftshift(J3);
B4=ifft2(K4);
%----visualizing the results----------------------------
------------------
figure(1)
imshow(I);colormap gray
title('original image','fontsize',14)
figure(2)
imshow(abs(A1),[-12 300000]), colormap gray
title('fft of original image','fontsize',14)
figure(3)
imshow(abs(B1),[12 290]), colormap gray
title('low pass filtered image','fontsize',14)
figure(4)
imshow(abs(B2),[12 290]), colormap gray
title('High pass filtered image','fontsize',14)
figure(5)
imshow(abs(B3),[12 290]), colormap gray
title('band pass filtered image','fontsize',14)
figure(6)
imshow(abs(B4),[12 290]), colormap gray
title('band stop filtered image','fontsize',14)
```

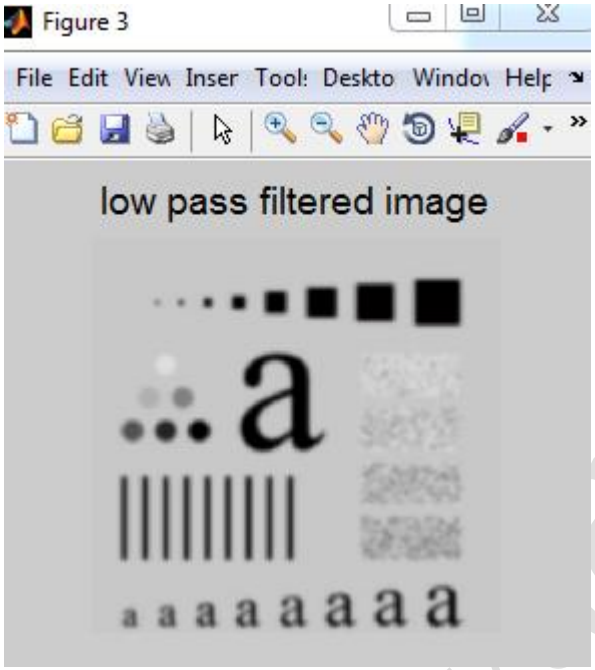


Figure 3
File Edit View Inser Tools Deskto Windov Help
low pass filtered image

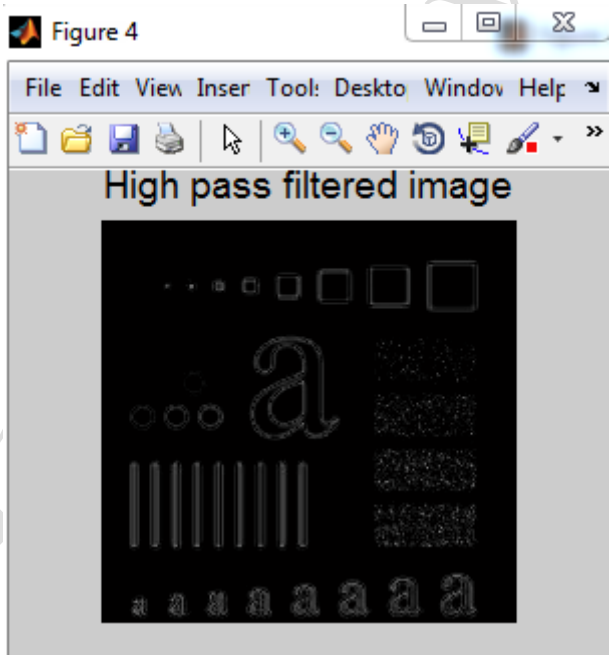


Figure 4
File Edit View Inser Tools Deskto Windov Help
High pass filtered image

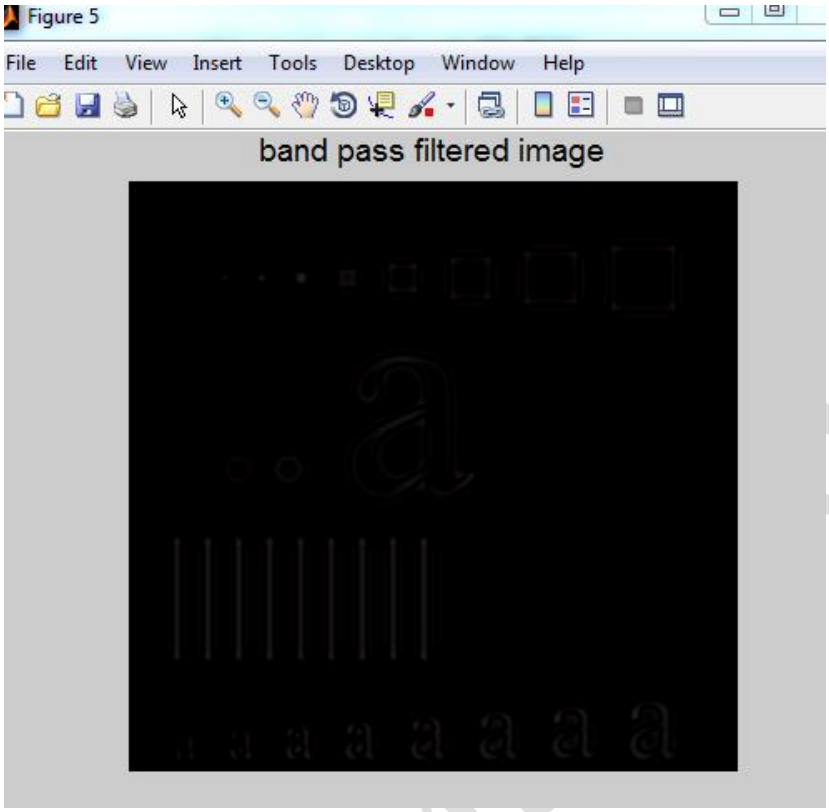


Figure 5
File Edit View Insert Tools Desktop Window Help
band pass filtered image

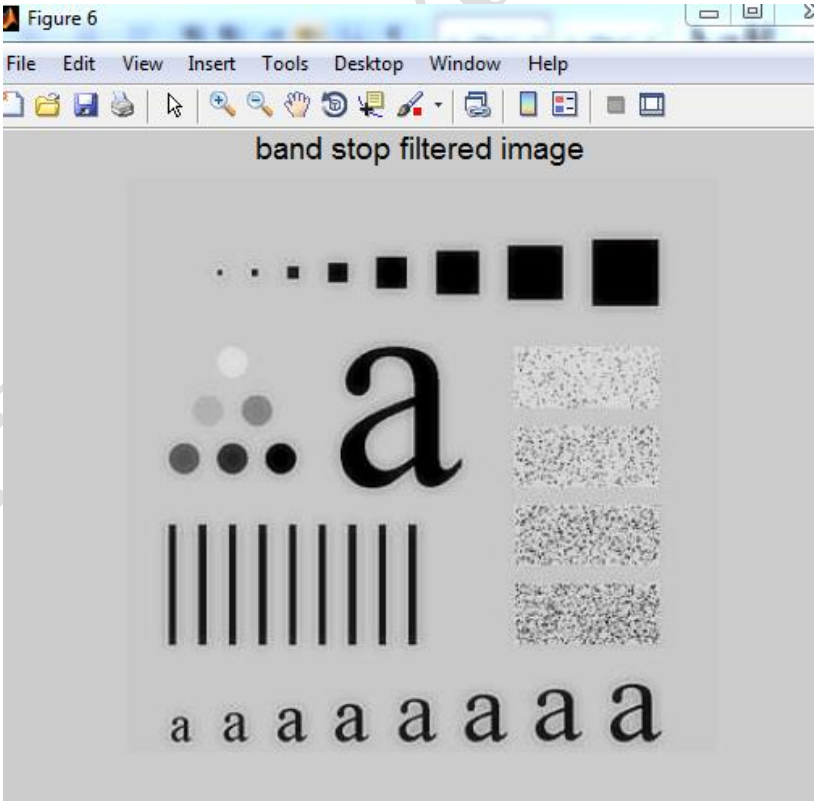
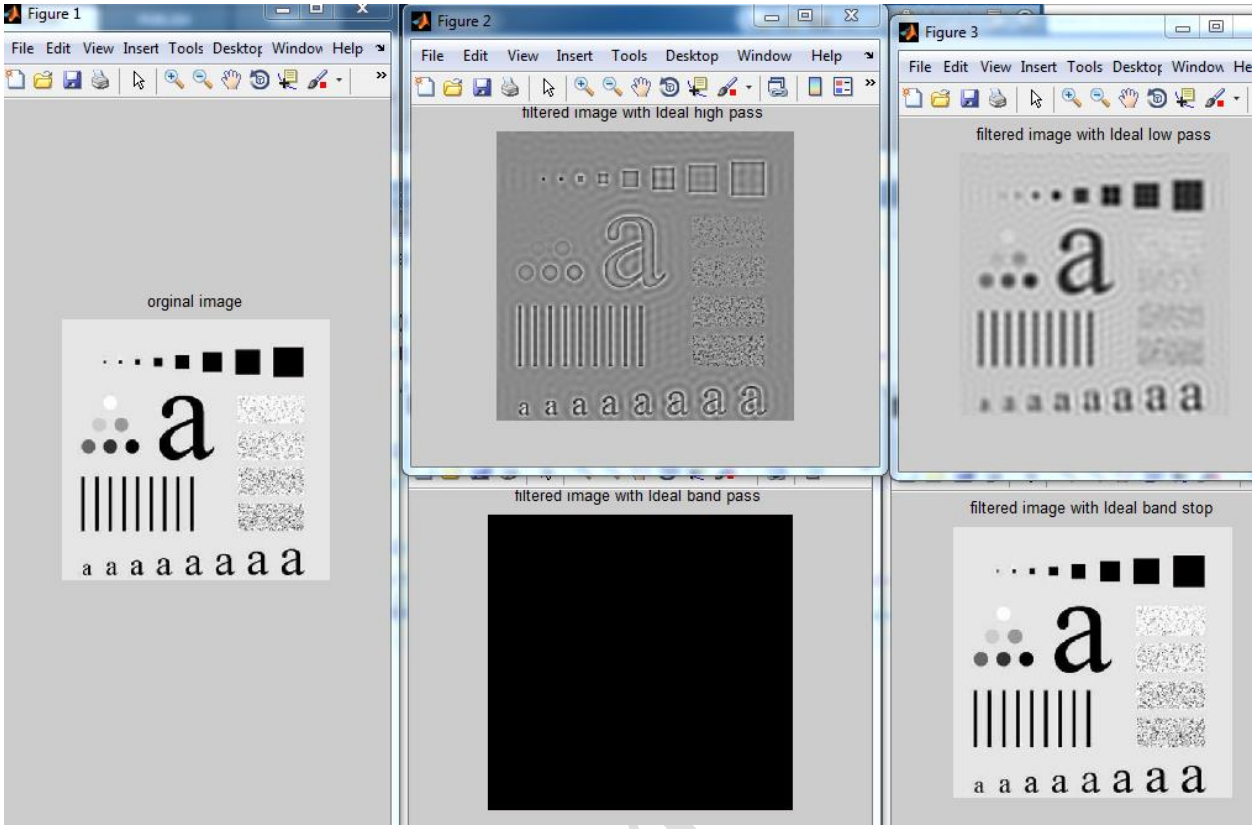


Figure 6
File Edit View Insert Tools Desktop Window Help
band stop filtered image

The Gaussian filter is also quieter than the Butterworth, and does not have the resonance and rotation of an ideal filter as seen in the figure.

Question 10) Demonstrate the ideal high-pass, low-pass and intermediate filter and band removal with a cut-off frequency of 30.

```
I=imread('C:\Users\Marjan\Documents\MATLAB\Marjan.jpg')
;
f=rgb2gray(I);تصویر خاکستری
[M,N]=size(f);سایز تصویر
F=fft2(double(f));تبدیل فوریه دوبعدی
u=0:(M-1);
v=0:(N-1);
idx=find(u>M/2);
u(idx)=u(idx)-M;
idy=find(v>N/2);
v(idy)=v(idy)-N;
[V,U]=meshgrid(v,u);
D=sqrt(U.^2+V.^2); فرمول ایده آل
P=30;
H=double(D>=P);
G=H.*F; بالاگذر فیلتر
g=real(ifft2(double(G)));
figure,imshow(f),title('orginal
image');figure,imshow(g,[ ]);title('filtered image with
Ideal high pass');
G=(1-H).*F;فیلتر پایین گذر
g=real(ifft2(double(G)));
figure,imshow(g,[ ]);title('filtered image with Ideal
low pass');
B=(1-H).*H; فیلتر میان گذر
G=B.*F;
g=real(ifft2(double(G)));
figure,imshow(g,[ ]);title('filtered image with Ideal
band pass');
G=(1-B).*F;  فیلتر حذف باند
g=real(ifft2(double(G)));
figure,imshow(g,[ ]);title('filtered image with Ideal
band stop');
```

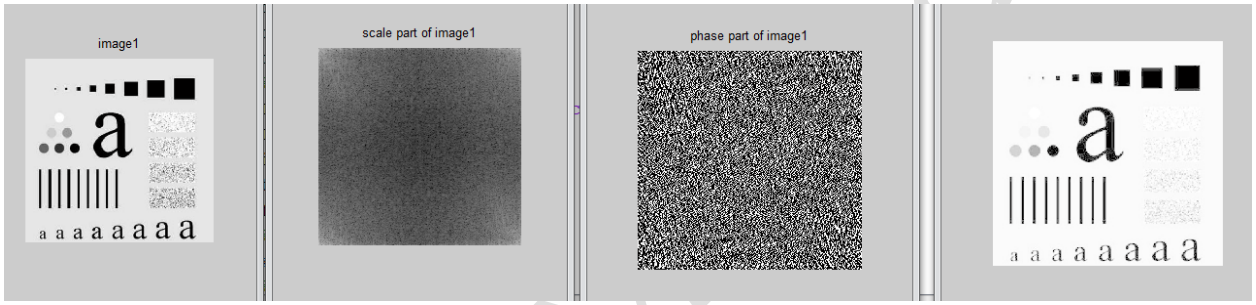


As we can see in the image of the ideal low-pass and high-pass, the rotating mode is very evident in the image.

Question 11) Display the Fourier transform phase and the Fourier transform size of the image, and then take the following image using the logarithmic function of the Fourier transform image.

```
clc;
clear;
close all;
%%%%%%%%%%%%%%%%%%%%%%
I1=imread('C:\Users\Marjan\Documents\MATLAB\test.jpg');
I1=rgb2gray(I1);
imshow(I1);
title('image1');
figure;
%%%%%%%%%%%%%%%%%%%%%%
x1=fft2(I1);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
r1=abs(x1);
imshow(log(r1+1),[]);
title('scale part of image1');
figure;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=angle(x1);
imshow(t);
title('phase part of image1');
figure;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
imshow(log(ifft2(x1)+1),[]);
```



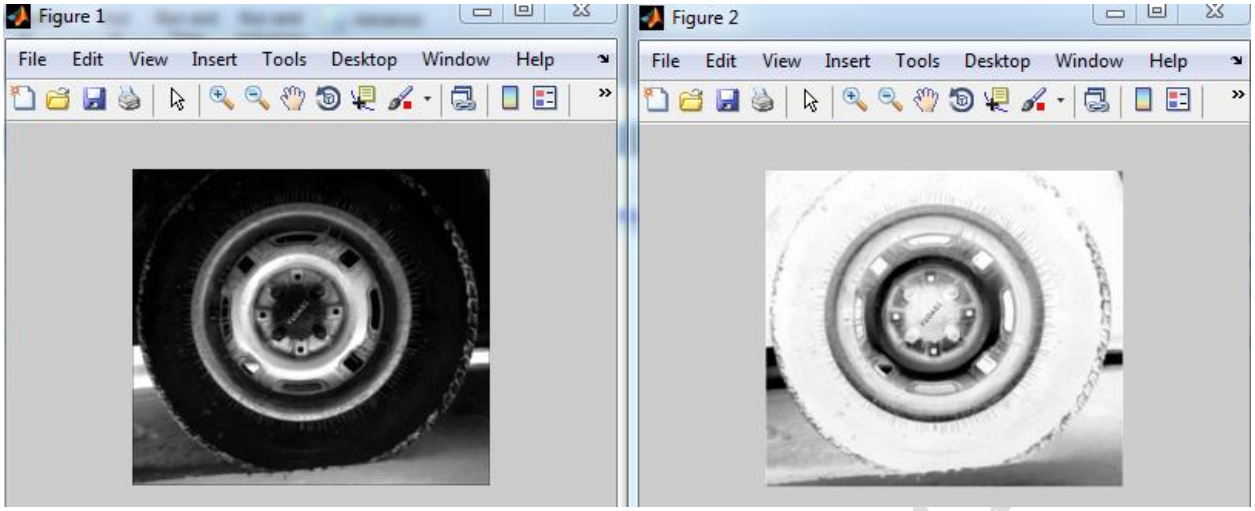
Convert the image from input to gray scal Using the fft2 function, we take a 2D Fourier transform image. Apply the size using the abs command. The angle angle function obtains for us. For the Fourier transform image, we use the ifft2 function, which we took from the logarithm function, which turned on the image.

Question 12) Display a negative of an image.

```
I=imread('tire.tif');خواندن تصویر
imshow(I)
J=imcomplement(I); نگاتیو تصویر
figure, imshow(J)نمایش تصویر
```

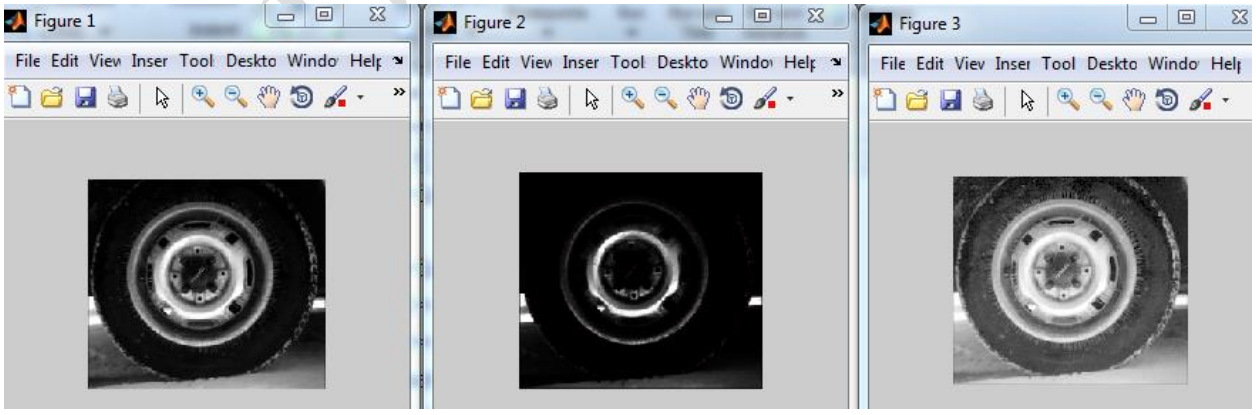
Black becomes white and white turns black. Between that, according to the maximum intensity (for example, if it is 255, it becomes 255-x) becomes the maximum intensity minus the current

intensity.



Question 13) Gamma Conversion Display an image with three different gammas of 1, 3 and 0.4, respectively.

```
I=imread('tire.tif');

J=imadjust(I,[],[],1); اعمال تبدیل گاما با مقدار 1
J2=imadjust(I,[],[],3); اعمال تبدیل گاما با مقدار 3
J3=imadjust(I,[],[],0.4); اعمال تبدیل گاما با مقدار 0.4
imshow(J); نمایش نتایج
figure,imshow(J2);
figure,imshow(J3);
```

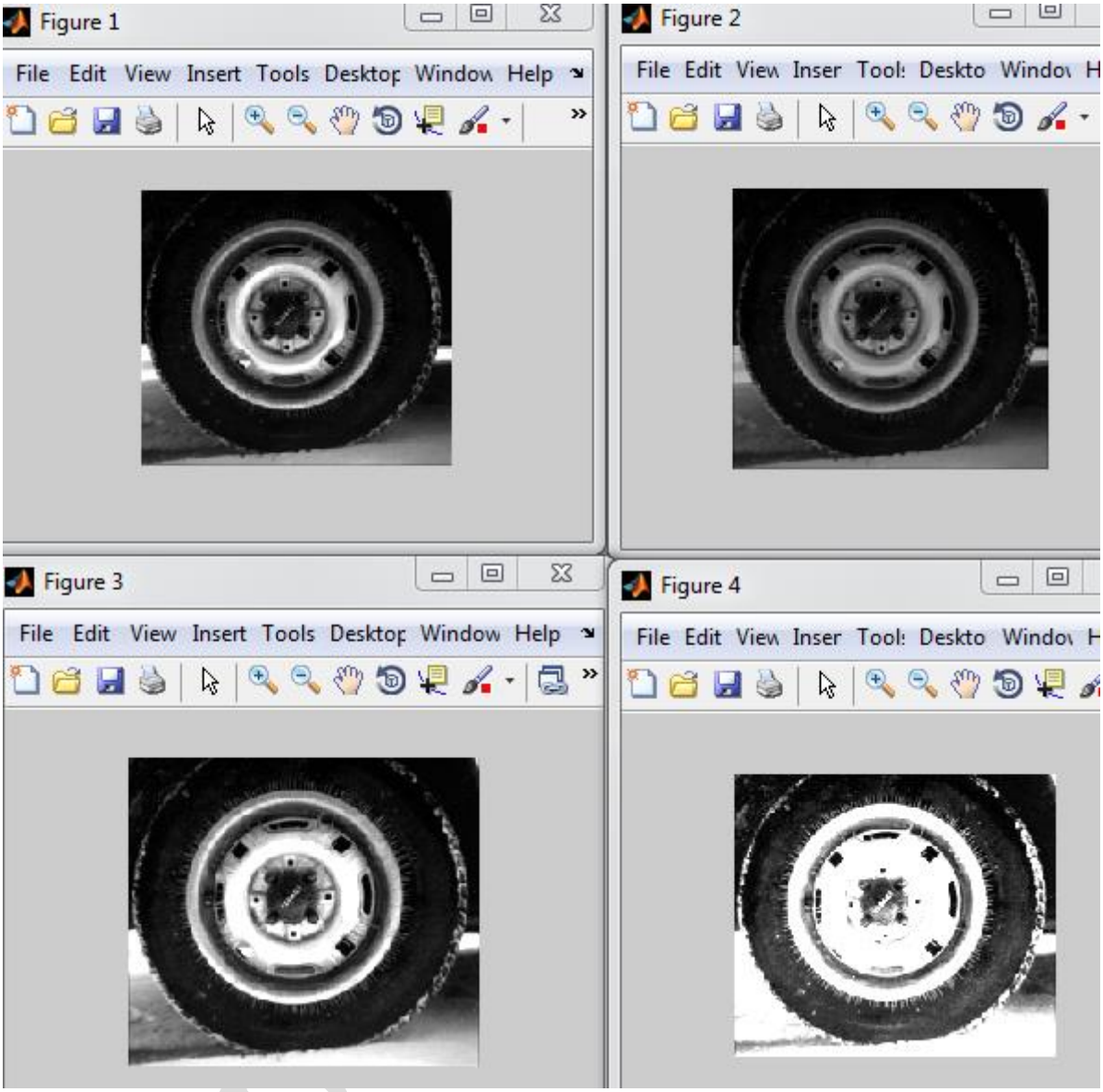


It was the same for value 1, for values greater than 1, such as 3, it becomes dark, the intensity decreases, and for values less than 1, it becomes 0.4, the image intensifies.

Question 14) Apply the logarithm conversion with coefficients 1, 2 and 5 on the image.

```
I=imread('tire.tif'); خواندن از ورودی
imshow(I) نمایش تصویر
I2=im2double(I); تبدیل به خاکستری
J=1*log(1+I2); تبدیل با ضرایب خواسته شده
J2=2*log(1+I2);
J3=5*log(1+I2);
figure, imshow(J) نمایش نتیجه
figure, imshow(J2)
figure, imshow(J3)
```

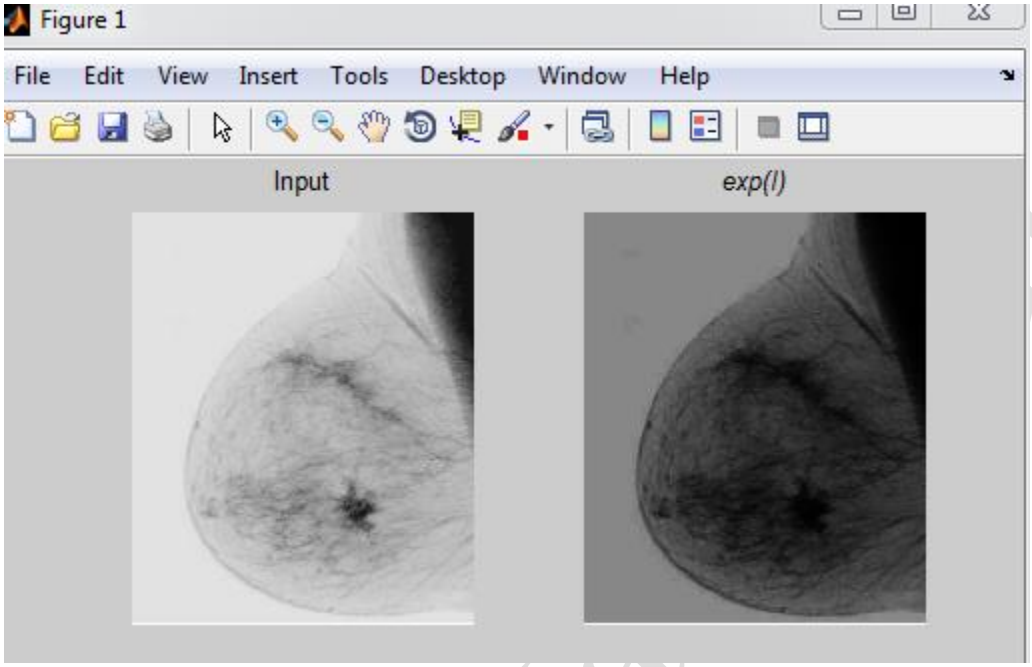
The following images from Figure 1 to 4 include the original image and the coefficient 1 in the image logarithm and the coefficients 2 and 5 in the image logarithm, respectively.



We see that as the logarithmic coefficient increases, the image becomes brighter.

Question 15) Implement the inverse logarithm on an image.

```
Figure;
L = 256; ماکزیمم سطح شدت
I = imread('C:\Users\Marjan\Documents\MATLAB\m1.jpg');
exp_I = uint8((exp(double(I)) .^ (log(L) / (L-1))) -
1); فرمول عکس لگاریتم
subplot(1, 2, 1); imshow(I); title('Input');تصویر ورودی
subplot(1, 2, 2); imshow(exp_I);
title('\itexp(I)');نتیجه
```



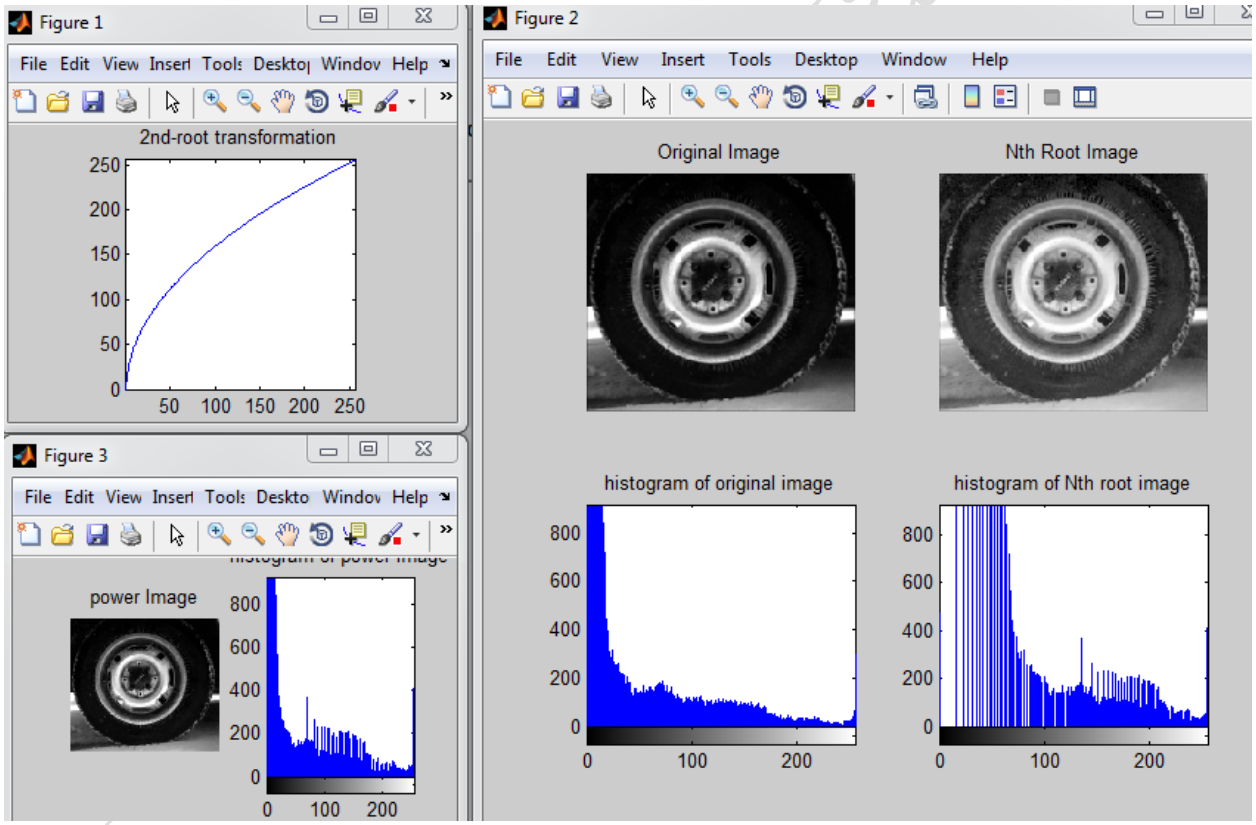
The logarithm image darkens the image. Reduces the intensity.

Question 16) Get the root and the second power of the image.

```
% تمام شکل های باز را ببندید و همه متغیرهای فضای کاری
را پاک کنید.
% یک تابع ریشه n که n ایجاد کنید است 2 برابر با.
close all ;clear;clc;
x = 0:255;n=2;c=255/(255 ^ n);
root = nthroot((x/c), n); ریشه
figure, plot(root),title('2nd-root transformation'),
axis tight, axis square
I = imread('tire.tif');
I_root = uint8(root(I + 1));
figure,subplot(2,2,1), imshow(I), title('Original
Image');خروجی
subplot(2,2,2), imshow(I_root), title('Nth Root
Image');
subplot(2,2,3),imhist(I),title('histogram of original
image');هیستگرام
```

```
subplot(2,2,4), imhist(I_root), title('histogram of Nth
root image');
                        %یک تابع تبدیل توان n را ایجاد کنید.
power=c*(x.^n); فرمول توان
figure, plot(power),title('2nd-power transformation');
axis tight, axis square خروجی
I_power = uint8(power(I_root + 1));
subplot(1,2,1), imshow(I_power), title('power Image');
subplot(1,2,2), imhist(I_power), title('histogram of
power image');
```

Lightens the root of the image and darkens the power of the image.

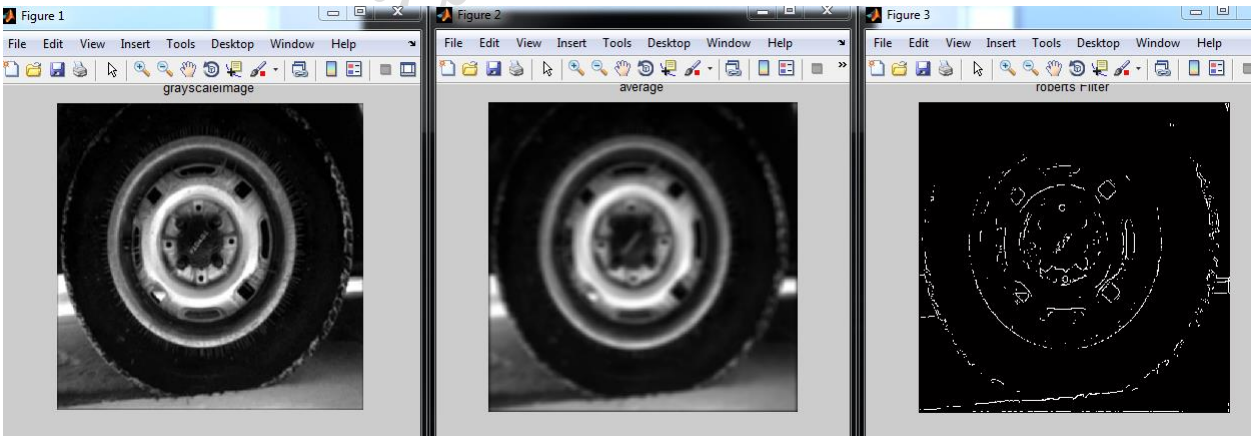


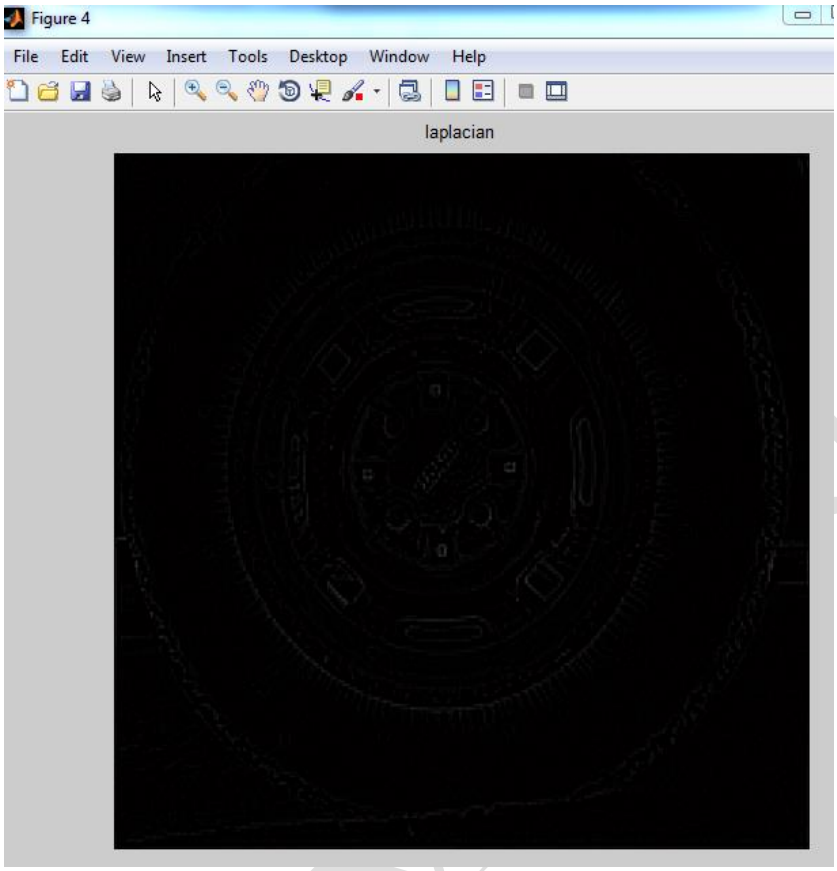
Question 17) Apply averaging Roberts, Laplacein and Gaussian filters to an image.

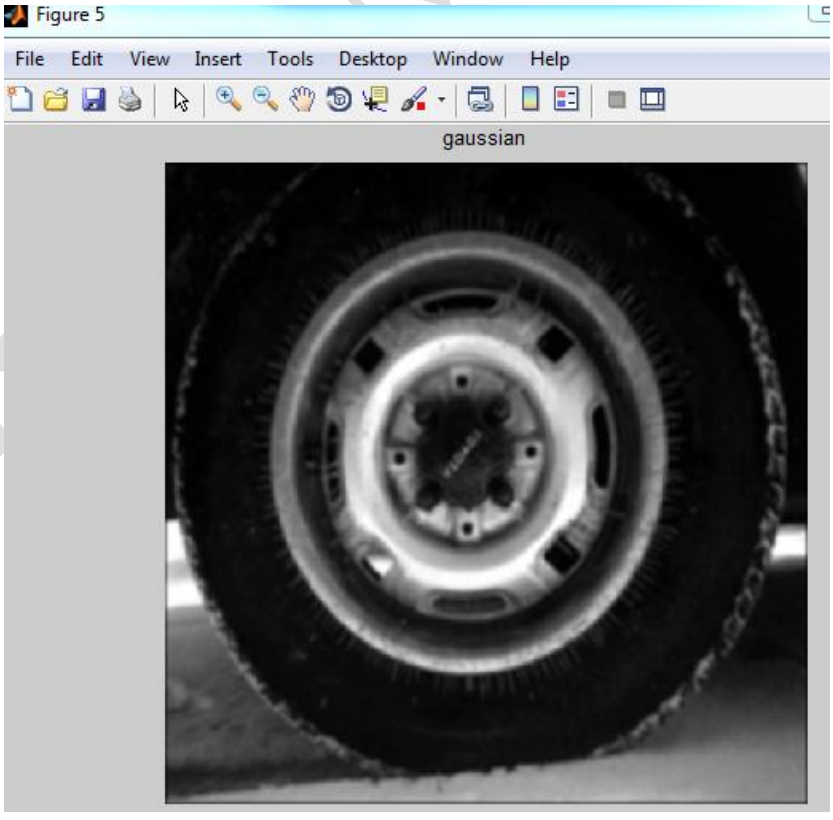
```
clc;
clear;
```

June 20, 2022

```
close all;
%%%%%%%%%%%%%%%%%%%%%%%%
% I=imread('tire.tif');
I=imread('C:\Users\Marjan\Documents\MATLAB\Marjan.jpg')
;
I=rgb2gray(I);
I=imresize(I,[512 512]);
imshow(I);title('grayscaleImage')

mask=fspecial('average',[9 11]); فیلتر میانگین گیری
I2=filter2(mask,I); اعمال فیلتر بر تصویر
figure;
imshow(uint8(I2));title('average')نمایش نتیجه

z = edge(I,'roberts');فیلتر روبرتز اعمال
figure,imshow(z);title('roberts Filter');

mask=fspecial('laplacian');لاپلاسین فیلتر
I2=filter2(mask,I);اعمال فیلتر
figure;
imshow(uint8(I2));title('laplacian') نتیجه

G = fspecial('gaussian',[5 5],1); فیلتر گوسی
Ig = imfilter(I,G,'same'); اعمال فیلتر بر تصویر
figure,imshow(Ig);title('gaussian')
```







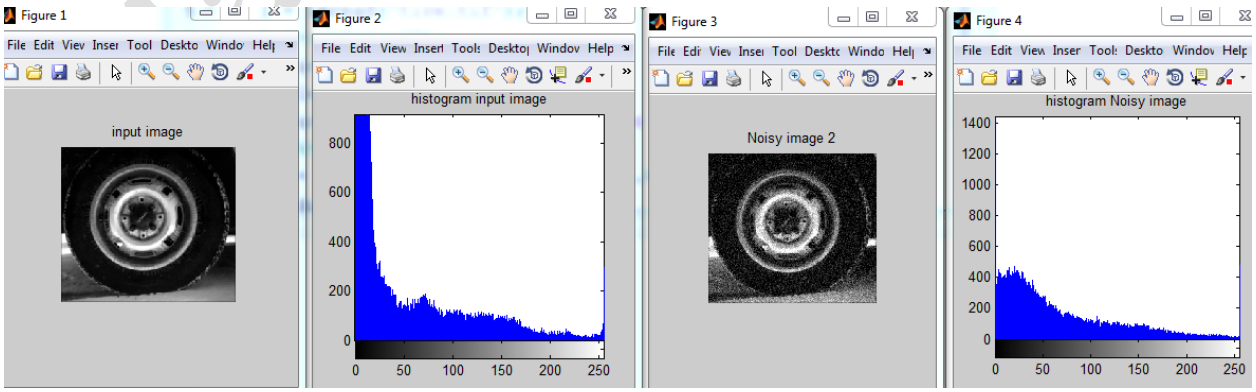
June 20, 2022

Averaging blurs and Gauss smoothes, but Laplacein and Roberts do the edge-finding.

Question 18) Take an image from the input and apply Gaussian noise to the image. And compare the original image histogram and the noise image.

```
clc
clear
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I =
imread('C:\Users\Marjan\Documents\MATLAB\Marjan.jpg');
I=rgb2gray(I); خاکستری کردن تصویر
figure,imshow(I);تصویر اصلی
title('input image');
figure,imhist(I);title('histogram input
image'); هیستگرام
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
J = imnoise(I,'gaussian',0.01,0.01);
     نویز گوسی با میانگین و واریانس 0.01 بر تصویر اعمال
                                                    میشه
 figure,imshow(J)تصویر نویزی
 title('Noisy image 2')
 figure,imhist(J);title('histogram Noisy image');
هیستگرام تصویر نویزی
```

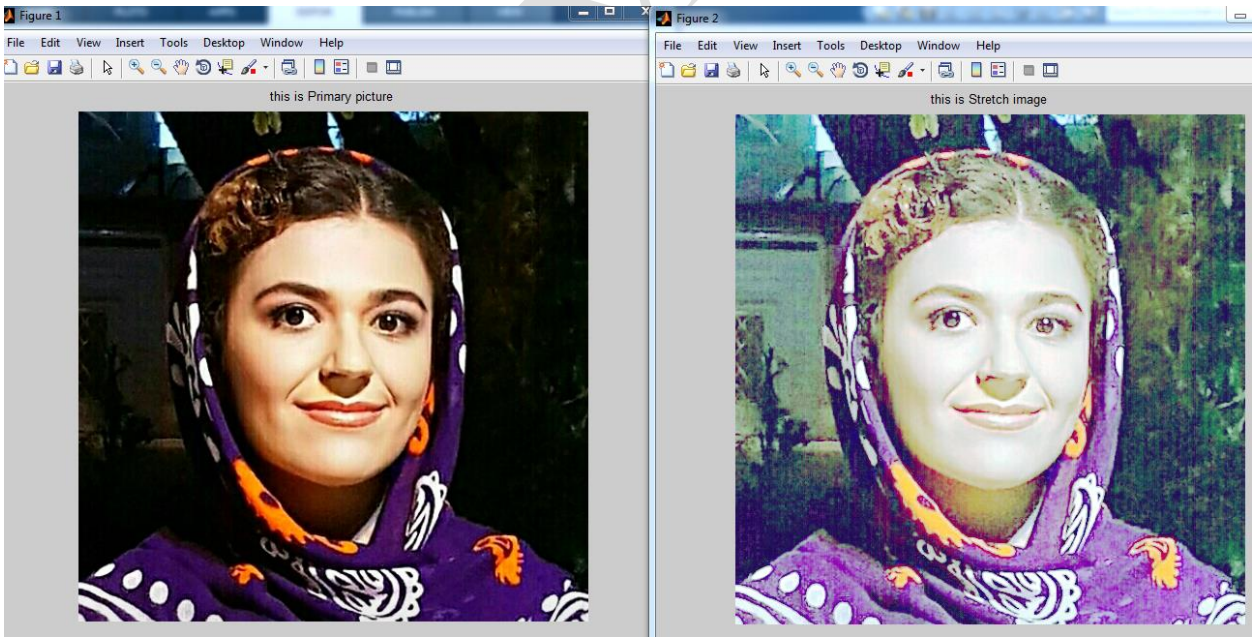
Take an image from the input and apply an amount of noise to the image using the imnoise command. And here we give the mean value and variance.



As can be seen from the histogram, the Gaussian noise image is clearly applied to the image.

Question 19) Take a color image of yourself and stretch the RGB each one separately and then put them on top of each other to display the result.

```
clc
clear
close all
img=imread('C:\Users\Marjan\Documents\MATLAB\Marjan.jpg
'); خواندن تصویر
figure,imshow(img);title('this is Primary picture');
R = img(:,:,1); کانال قرمز
G = img(:,:,2); کانال سبز
B = img(:,:,3); کانال آبی
eqimg(:,:,1) = histeq(R); تعدیل صفحه قرمز
eqimg(:,:,2) = histeq(G); تعدیل صفحه سبز
eqimg(:,:,3) = histeq(B); تعدیل صفحه آبی
figure,imshow(eqimg);title('this is Stretch image'); روی
گذاشتن کانال ها و نمایش نتیجه
```

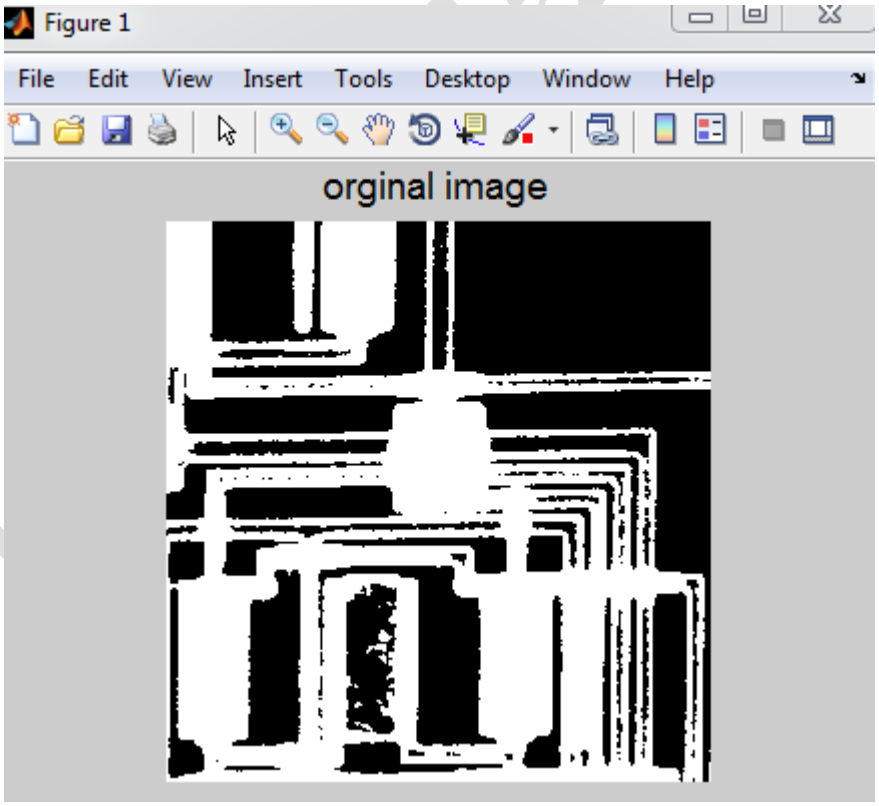


We see that modifying three separate red, green, and blue pages from an image and their consensus did not improve the color of the image, but rather ruined it.

Question 20) Apply erosion and expansion on an image with a structural element within a radius of 2.

```
BW1 = imread('circbw.tif'); خواندن تصویر
figure(1);imshow(BW1);title('orginal
image','fontsize',14) عنوان نتیجه با فونت 14 نمایش بده
SE = strel('disk',2); عنصر ساختاری با دیسک به شعاع 2
پیکسل.
BW2 = imerode(BW1,SE); فرسایش با عنصر ساختاری
figure(2);imshow(BW2);title('Erosion
image','fontsize',14) نتیجه فرسایش
BW3 = imdilate(BW1,SE); گسترش با عنصر ساختاری
figure(3);imshow(BW3);title('Dilation
image','fontsize',14) نتیجه گسترش
```

Original image and erosion and expansion



Erosion darkens the image and lightens the expansion of the image, and in this first form, the erosion makes objects thinner and shorter, and fills in the gaps.

June 20, 2022

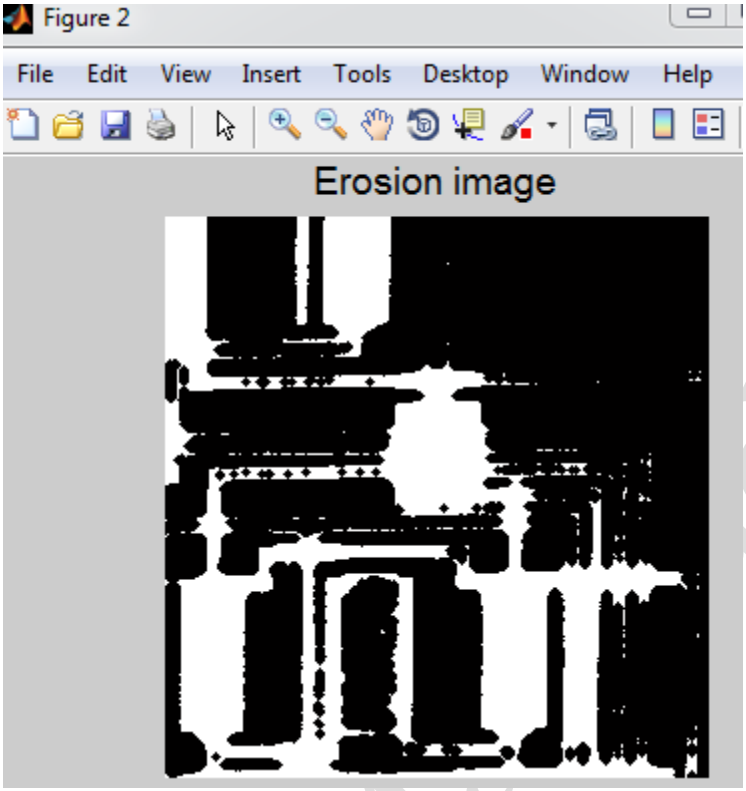


Figure 2
File Edit View Insert Tools Desktop Window Help
Erosion image

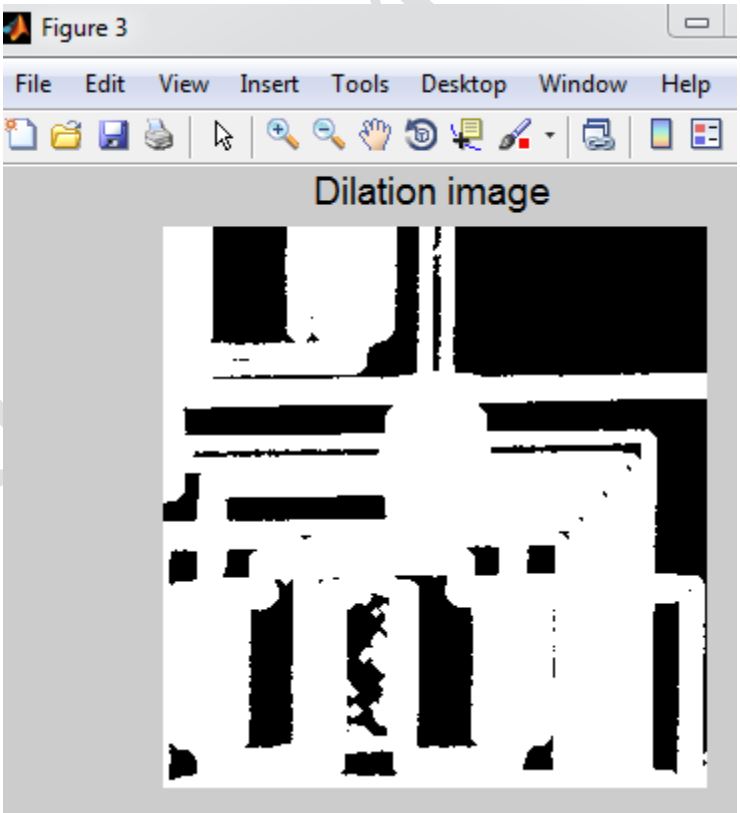


Figure 3
File Edit View Insert Tools Desktop Window Help
Dilation image

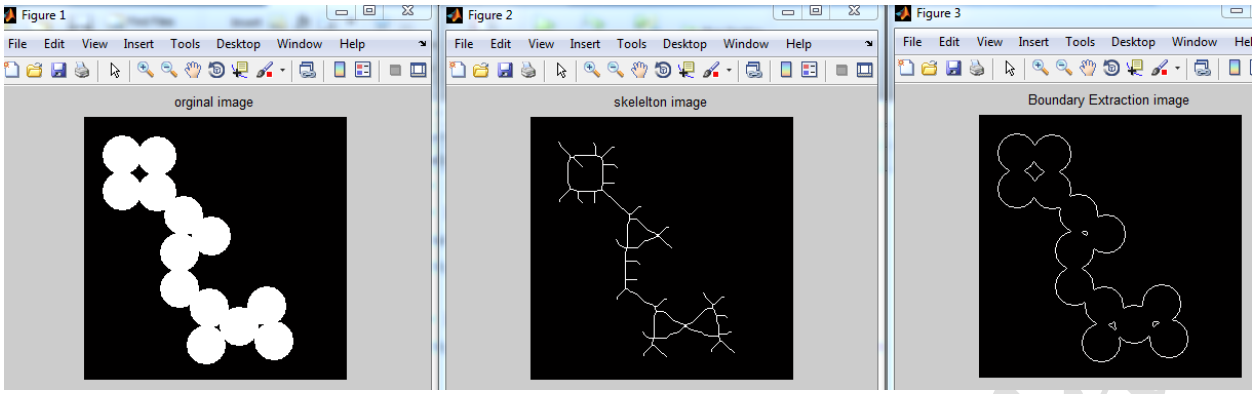
Question 21) Apply the opening to an image with a structural element within a radius of 2 and then apply the close to the result.

```
BW1 =
imread('C:\Users\Marjan\Documents\MATLAB\se.jpg');
figure(1);imshow(BW1);title('orginal
image','fontsize',14)
SE = strel('disk',2);عنصر ساختاری
BW2 = imopen(BW1,SE);   باز کردن تصویر با عنصر ساختاری
figure(2);imshow(BW2);title('open image','fontsize',14)
BW3 = imclose(BW2,SE);  بستن تصویر با عنصر ساختاری
figure(3);imshow(BW3);title('imclose
image','fontsize',14)
```

Opening action removes noise less than 2 pixels, but can cause fingerprint fractures. After one closing operation, we do not see much change in the image because closing does not guarantee filling the fractures, but it may help a little. Did.

Question 22) Get the framework and extract the border for the image.

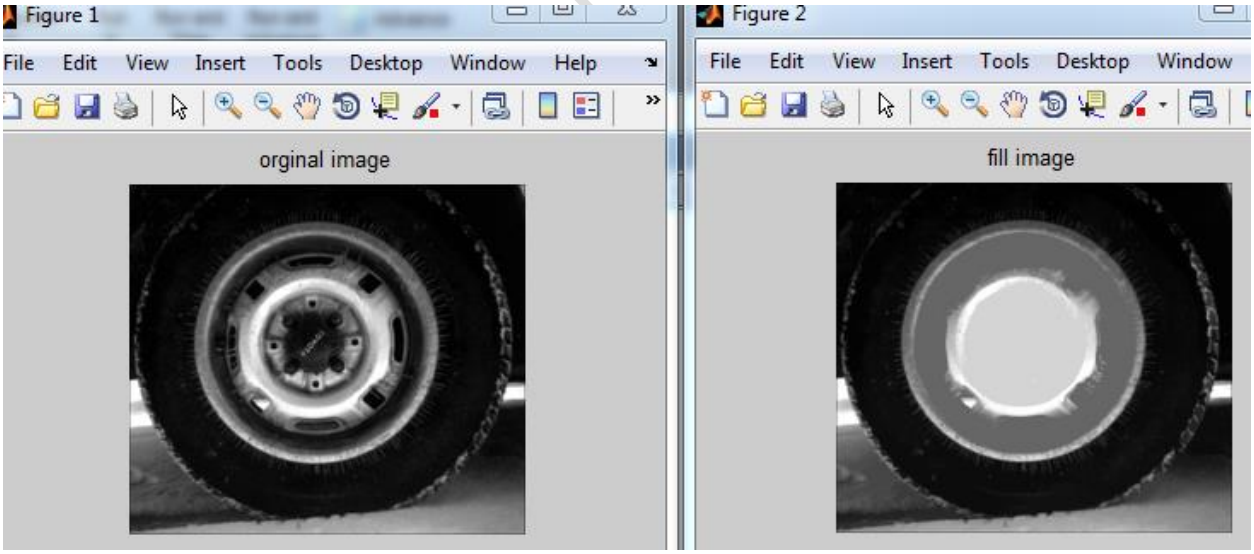
```
BW1 = imread('circles.png');
BW2 = bwmorph(BW1,'skel',inf); چهارچوب
imshow(BW1);title('orginal image');
figure,imshow(BW2);title('skelelton image');

BW3 = bwmorph(BW1,'remove'); استخراج مرز
figure,imshow(BW3);title('Boundary Extraction image');
```



It has framed the framework and extracted the boundaries of the shape.

Question 23) Apply the hole filling on the image.

```
I = imread('tire.tif');
I2 = imfill(I); پر کردن
figure, imshow(I),title('orginal image');
figure, imshow(I2),title('fill image');
```

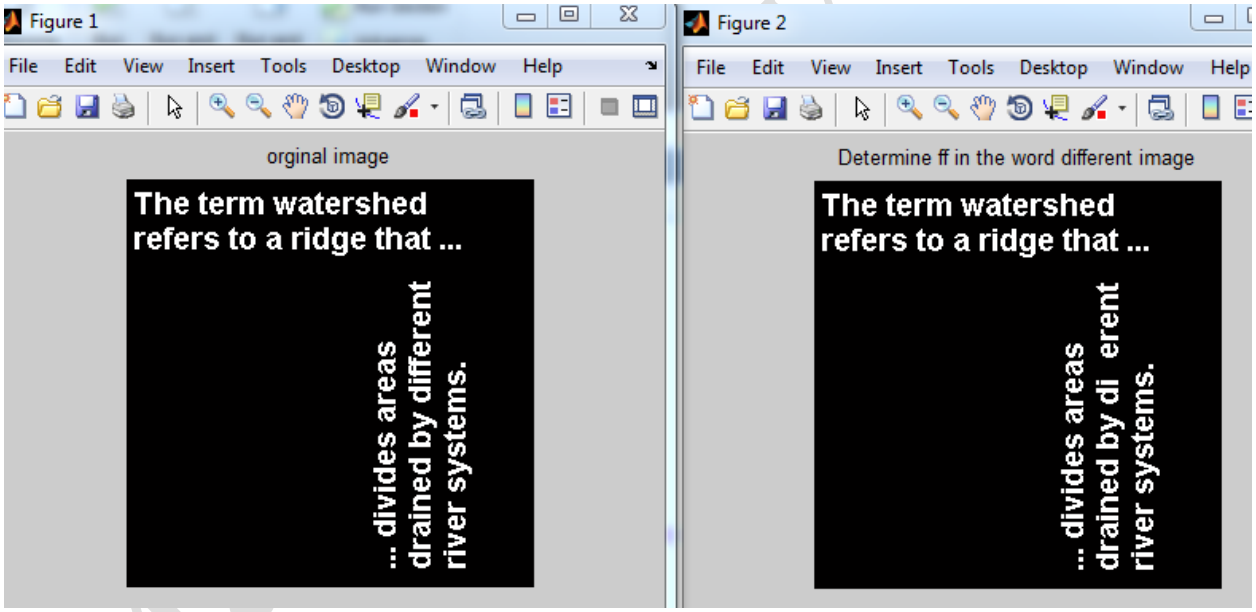


We see that the tire layers are each filled and this image conveys the concept of filling very nicely.

Question 24) Find and remove the largest connected component.

```
BW = imread('text.png');
imshow(BW);title('orginal image');
CC = bwconncomp(BW); تعداد اجزای متصل شده را در تصویر پیدا کنید.
numPixels = cellfun(@numel,CC.PixelIdxList);
تعیین کنید که کدام جزء بزرگ ترین در تصویر است و آن را پاک کنید (همه پیکسل ها را روی 0 قرار دهید).
[biggest,idx] = max(numPixels);
BW(CC.PixelIdxList{idx}) = 0;
figure;
imshow(BW);title('Determine ff in the word different image');
```

Display the image, note that it happens to be the largest component of two consecutive fs in the word **different**.



CC =

Connectivity: 8

[ImageSize: [256 256

NumObjects: 88

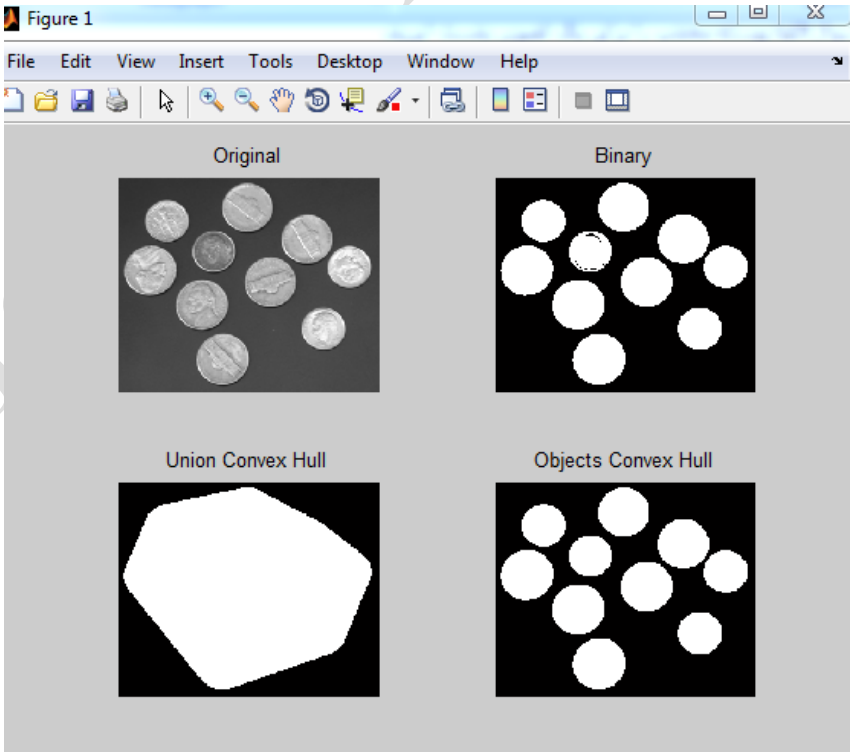
{PixelIdxList: {1x88 cell

The largest connected component is **ff** in **different**, which has been removed.

June 20, 2022

Question 25) Get a convex shell for an image.

```
subplot(2,2,1);
I = imread('coins.png');
imshow(I);
title('Original');

subplot(2,2,2);
BW = I > 100; آستانه تبدیل به باینری
imshow(BW);
title('Binary');

subplot(2,2,3);
CH = bwconvhull(BW); ناحیه پوسته محدب
imshow(CH);
title('Union Convex Hull');

subplot(2,2,4);
CH_objects = bwconvhull(BW,'objects');اشیا پوسته محدب
imshow(CH_objects);
title('Objects Convex Hull');
```

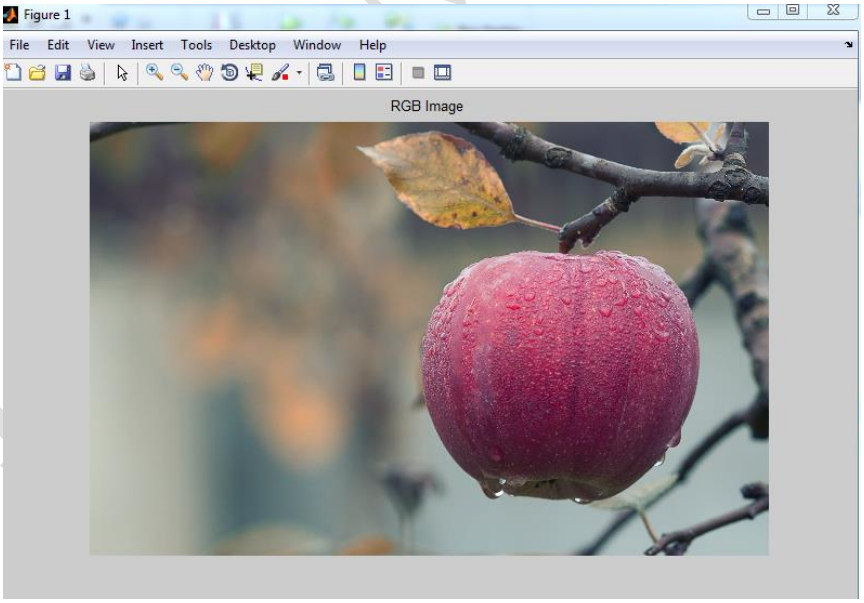


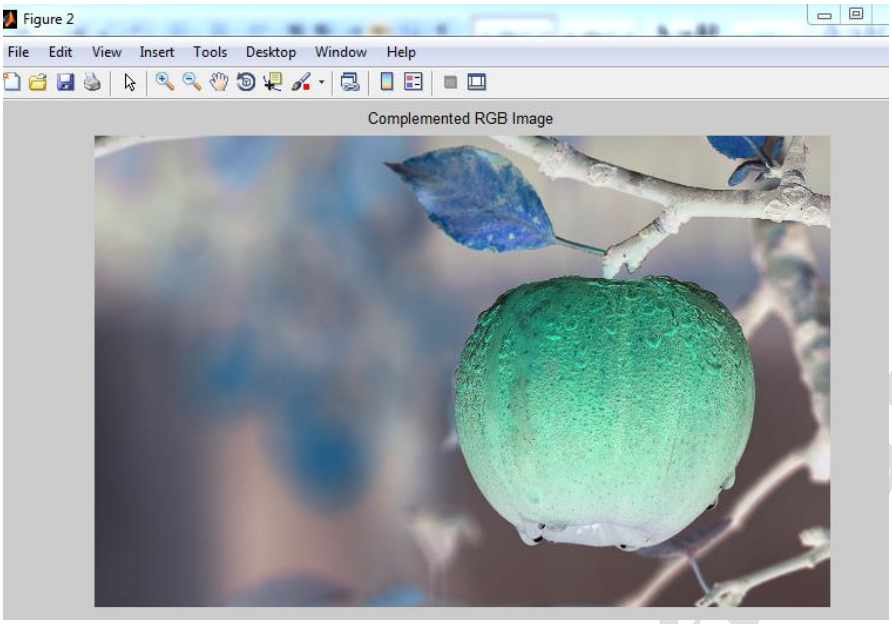
First the image is binary and then the convex shell in Figure 3 shows the areas very clearly and Figure 4 shows the convex shell objects.

Question 26) Get a complement to an rgb color image in MATLAB?

```
% در محیط متلب خواندن یک تصویر RGB
img=imread('C:\Users\Marjan\Documents\MATLAB\apple.jpeg
');
figure;imshow(img);title('RGB Image');

% تبدیل کلاس تصویر RGB به 'uint8'
img=im2uint8(img);

% هر پیکسل را با کم کردن آن از 255 تکمیل می کند
comp=255-img;

% نمایش تصویر RGB تکمیل شده
figure;imshow(comp);title('Complemented RGB Image');
```

خروجی:



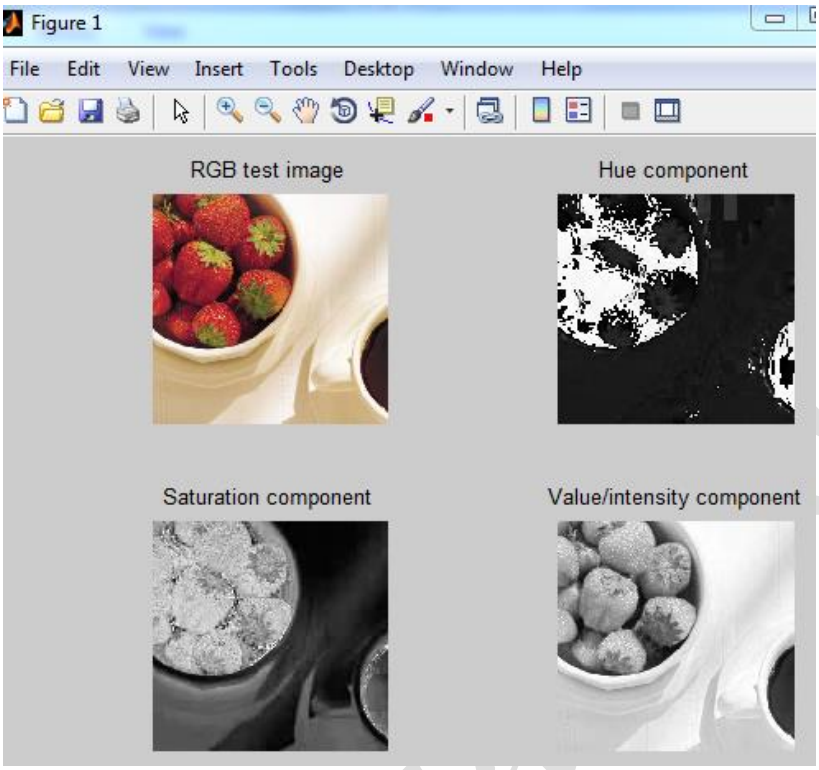


We see the complement of the color image which is the color complement according to Figure 6-32.

Question 27) Take an rgb color image to hsi space and display the image components?

```
testImg=imread('C:\Users\Marjan\Documents\MATLAB\image.
png');
                                        %تبدیل از RGB به HSI
hsvTestImg = rgb2hsv(testImg);
% Segregate hue, sat و value/intensity
hueTestImg = hsvTestImg(:, :, 1);
satTestImg = hsvTestImg(:, :, 2);
valTestImg = hsvTestImg(:, :, 3);
                    RGBو اجزای HSI مربوطه را نمایش دهید
figure; subplot(2, 2, 1)
imshow(testImg); title('RGB test image');
subplot(2, 2, 2)
imshow(hueTestImg); title('Hue component');
subplot(2, 2, 3)
imshow(satTestImg); title('Saturation component');
subplot(2, 2, 4)
imshow(valTestImg); title('Value/intensity component');
```

خروجی:

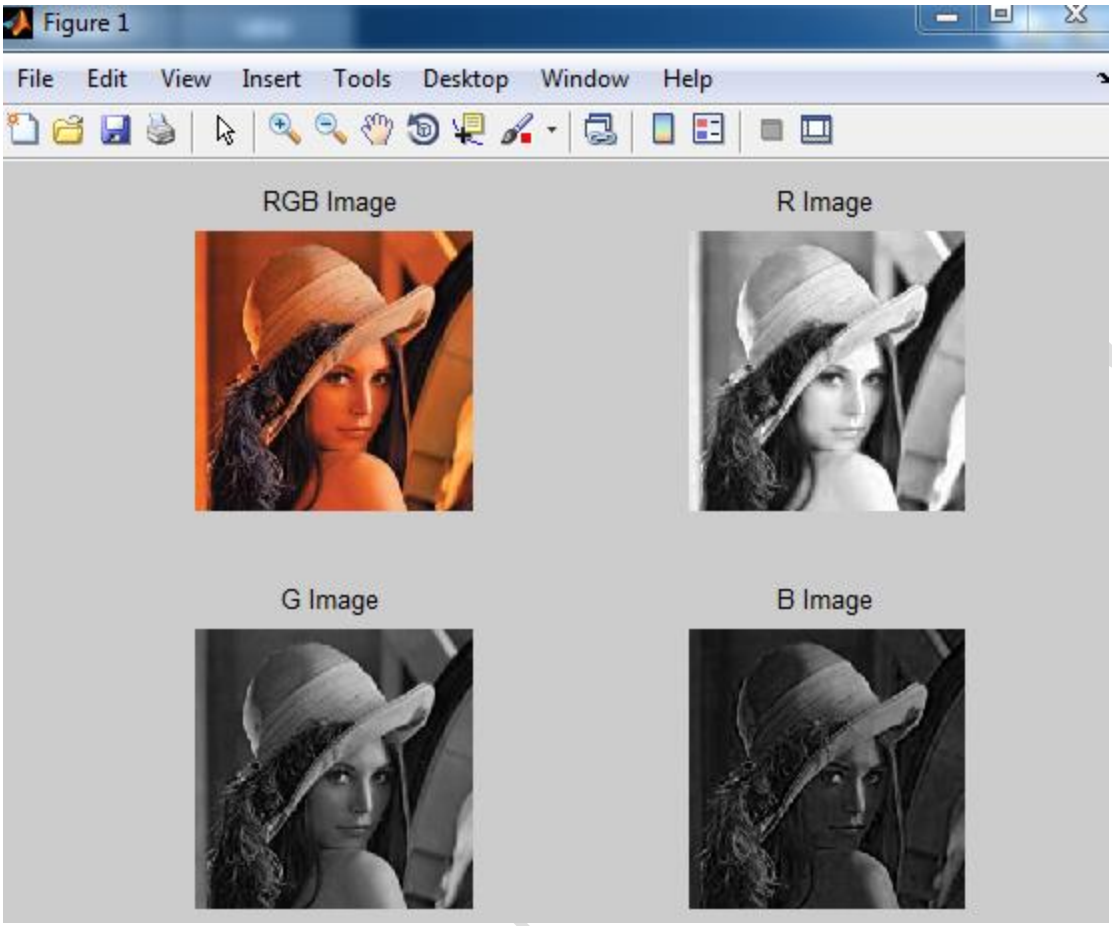


We see curtains of color and saturation and intensity.

Question 28) Display the rgb components of the image separately?

```
Img=imread('C:\Users\Marjan\Documents\MATLAB\lena.jpg')
;
figure; subplot(2, 2, 1)
imshow(Img);title('RGB Image');
R=Img(:, :, 1); کانال قرمز
G=Img(:, :, 2); کانال سبز
B=Img(:, :, 3); کانال آبی
subplot(2, 2, 2);imshow(R);title('R Image');نمایش کانال
subplot(2, 2, 3);imshow(G);title('G Image');
subplot(2, 2, 4);imshow(B);title('B Image');
```

خروجی:



Any image that is darker means that the color is less in the original image and vice versa, each one is lighter, ie more in the original image.

Question 29) Pseudo-color image processing Perform light intensity segmentation on a gray image.

Here is dedicated to a range of gray surfaces of one color

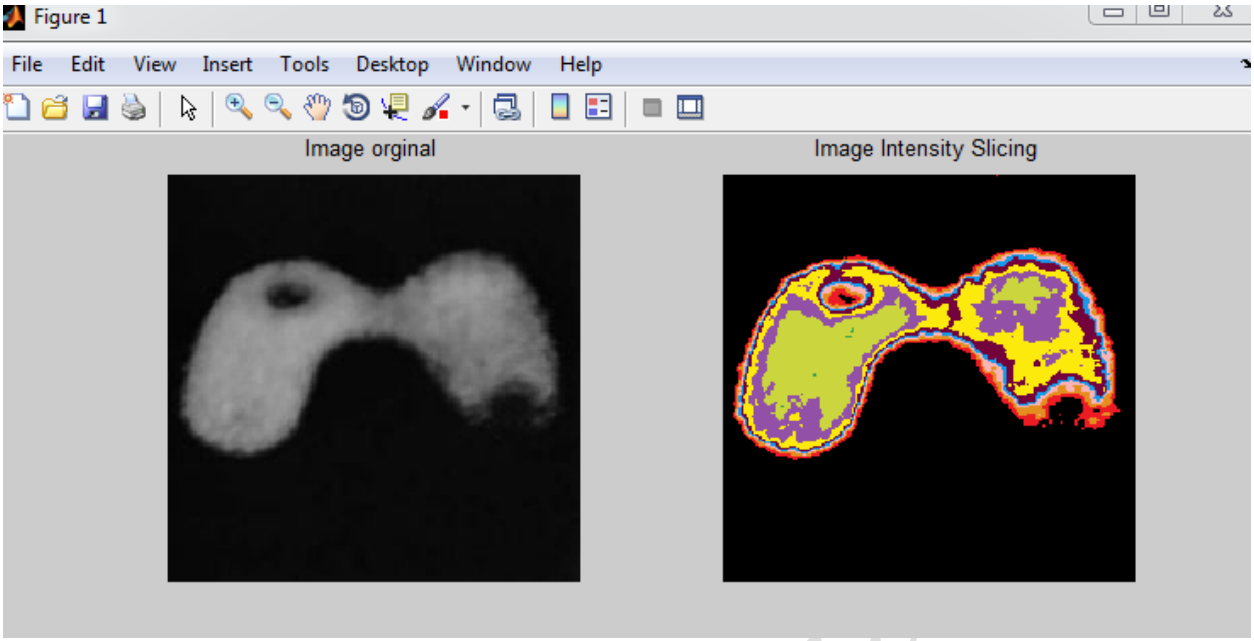
```
clc;clear all;close all;
y=imread('C:\Users\Marjan\Documents\MATLAB\29\image.jpg
');
%
y=imread('C:\Users\Marjan\Documents\MATLAB\Marjan.jpg')
;
y=rgb2gray(y); تبدیل تصویر به خاکستری
[p,q,r]=size(y);
```

Image pixels are divided into 14 sections based on their intensity, and each section is assigned a color.

```
For i=1:1:p
    for j=1:1:q
        if (y(I,j)>= 0) && (y(I,j)< 18)
            x(I,j,1)=0;
            x(I,j,2)=0;
            x(I,j,3)=0;
        elseif (y(I,j)>= 18) && (y(I,j)< 36)
            x(I,j,1)=237;
            x(I,j,2)=27;
            x(I,j,3)=36;
        elseif (y(I,j)>= 36) && (y(I,j)< 54)
            x(I,j,1)=228;
            x(I,j,2)=142;
            x(I,j,3)=31;
        elseif (y(I,j)>= 54) && (y(I,j)< 72)
            x(I,j,1)=251;
            x(I,j,2)=179;
            x(I,j,3)=180;
        elseif (y(I,j)>= 72) && (y(I,j)< 90)
            x(I,j,1)=21;
            x(I,j,2)=154;
            x(I,j,3)=233;
        elseif (y(I,j)>= 90) && (y(I,j)< 108)
            x(I,j,1)=116;
            x(I,j,2)=3;
            x(I,j,3)=59;
        elseif (y(I,j)>= 108) && (y(I,j)< 126)
            x(I,j,1)=252;
            x(I,j,2)=234;
            x(I,j,3)=12;
        elseif (y(I,j)>= 126) && (y(I,j)< 144)
             x(I,j,1)=146;
            x(I,j,2)=80;
            x(I,j,3)=167;
        elseif (y(I,j)>= 144) && (y(I,j)< 162)
            x(I,j,1)=203;
            x(I,j,2)=213;
            x(I,j,3)=62;
```

```
        elseif (y(I,j)>= 162) && (y(I,j)< 180)
            x(I,j,1)=59;
            x(I,j,2)=165;
            x(I,j,3)=77;
        elseif (y(I,j)>= 180) && (y(I,j)< 198)
            x(I,j,1)=48;
            x(I,j,2)=85;
            x(I,j,3)=173;
        elseif (y(I,j)>= 198) && (y(I,j)< 216)
            x(I,j,1)=126;
            x(I,j,2)=180;
            x(I,j,3)=67;
        elseif (y(I,j)>= 216) && (y(I,j)< 232)
            x(I,j,1)=16;
            x(I,j,2)=233;
            x(I,j,3)=59;
        elseif (y(I,j)>= 232) && (y(I,j)< 255)
            x(I,j,1)=255;
            x(I,j,2)=255;
            x(I,j,3)=100;
        end
    end
end
subplot(1,2,1);
imshow(y);
title('Image orginal');
subplot(1,2,2);
x=x/255;
% image(x);
imshow(x);
title('Image Intensity Slicing');
```

خروجی:

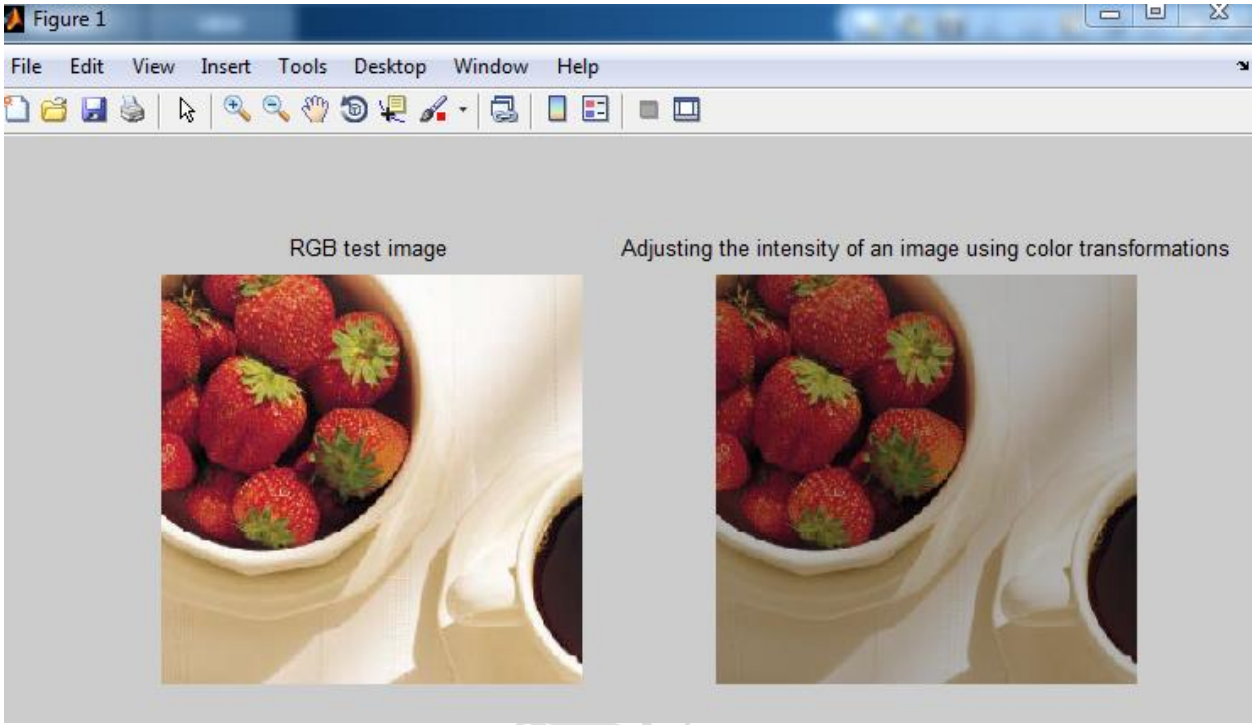


As we can see, with 14 colors, the intensity zones are segmented in the gray image and turned into a color image.

Question 30) Increase the color intensity by 30% in a color image.

```
%
testImg=imread('C:\Users\Marjan\Documents\MATLAB\image.
png');
testImg=imread('C:\Users\Marjan\Documents\MATLAB\30\Mar
jan.jpg');

wImg(:,:,1)= testImg(:,:,1)*0.7;
wImg(:,:,2)= testImg(:,:,2)*0.7;
wImg(:,:,3)= testImg(:,:,3)*0.7;

The image intensity in each color channel is multiplied
by 0.7, as in the example of 6-31 books

figure; subplot(1, 2, 1)
imshow(testImg); title('RGB test image');
subplot(1, 2, 2)
imshow(wImg); title('Adjusting the intensity of an
image using color transformations');
```

Because the intensity of all channels is multiplied by a number less than one, the image is darkened.

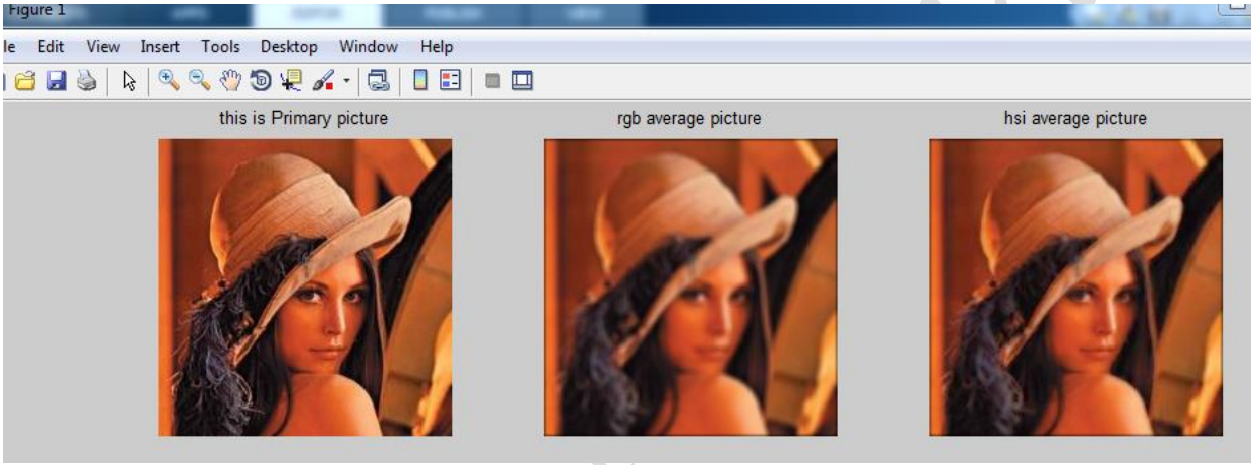


31) Image smoothing with 5 in 5 interpolation filter for rgb and hsi.

```
%
img=imread('C:\Users\Marjan\Documents\MATLAB\30\lena.jpg');
img=imread('C:\Users\Marjan\Documents\MATLAB\30\Marjan.jpg');
figure,subplot(1,3,1);
imshow(img);title('this is Primary picture');
C=fspecial('average',[5,5]); فیلتر متوسط 5x5
R = img(:,:,1);
G = img(:,:,2);
B = img(:,:,3);
averageimg(:,:,1) = imfilter(R,C); اعمال فیلتر در هر کانال
averageimg(:,:,2) = imfilter(G,C);
averageimg(:,:,3) = imfilter(B,C);
```

```
subplot(1,3,2);
imshow(averageimg);title('rgb average picture');نـتـیـجه
hsvTestImg = rgb2hsv(img); hsi تـبـدیـل rgb بـه
f = hsvTestImg(:, :, 3); hsi مـولـفـه شدت
hsvTestImg(:, :, 3) = imfilter(f,C);اعـمـال فـیـلـتـر
h=hsv2rgb(hsvTestImg); rgb بـه hsi تـبـدیـل
subplot(1,3,3);
imshow(h);title('hsi average picture');
```

خروجی:



As we can see, the interpolation in these two color conversions does not give the same result.

32) Laplacein rgb and hsi and the difference between them.

```
clc;
close all;
img=imread('C:\Users\Marjan\Documents\MATLAB\30\lena.jpg');
%
img=imread('C:\Users\Marjan\Documents\MATLAB\30\Marjan.jpg');
figure,subplot(1,4,1);
imshow(img);title('this is Primary picture');
```

```
R = img(:,:,1); کانال های رنگی
G = img(:,:,2);
B = img(:,:,3);
lap = [-1 -1 -1; -1 8 -1; -1 -1 -1]; فیلتر لاپلاسین
                                    اعمال فیلتر در هر کانال
resp(:,:,1) = imfilter(R, lap, 'conv');
resp(:,:,2) = imfilter(G, lap, 'conv');
resp(:,:,3) = imfilter(B, lap, 'conv');
نرمالسازی پاسخ
minR = min(resp(:));
maxR = max(resp(:));
resp = (resp - minR) / (maxR - minR);
افزودن به تصویر اصلی
sharpened = img + resp;
subplot(1,4,2);
imshow(sharpened);title('rgb lap picture');
 نمایشش
hsvTestImg = rgb2hsv(img);
                                    rgb به hsi تبدیل کن.
f = hsvTestImg(:, :, 3); مولفه شدت
hsvTestImg(:, :, 3) = imfilter(f , lap, 'conv'); اعمال
 فیلتر به آن
resp2=hsvTestImg(:, :, 3);
نرمالسازی پاسخ
minR = min(resp2(:));
maxR = max(resp2(:));
resp2 = (resp2 - minR) / (maxR - minR);
                                    تصویر اول را به hsi برده
Img1 = rgb2hsv(img);
resp2(:, :, 3) = Img1(:, :, 3) + resp2; افزودن به شدت
تصویر اصلی برای دومی و پرده رنگ و اشباع کاری نداریم.
resp2(:, :, 2)=hsvTestImg(:, :, 2);
resp2(:, :, 1)=hsvTestImg(:, :, 1);
                                    hsi به rgb تبدیل می کنیم.
sharpened2=hsv2rgb(resp2);
subplot(1,4,3);
imshow(sharpened2);title('hsi lap picture'); نمایش
نتیجه
   هر دو را به hsv برده اختلاف شدت شان میگیریم و قدرمطلق
میگیریم.
```

```
hsv1 = rgb2hsv(sharpened);
hsv2 = rgb2hsv(sharpened2);
diff=imabsdiff(hsv1(:, :, 3),hsv2(:, :, 3));


subplot(1,4,4);
imshow(diff);title(' - lap picture'); نـتـيـجـه اخـتـلاف
```

خروجی:

نتیجه اختلاف در تصویر مشهود است.

33) Segmentation in hsi color space based on red color in the left area of the image below. (Segmentation in hsi color space)

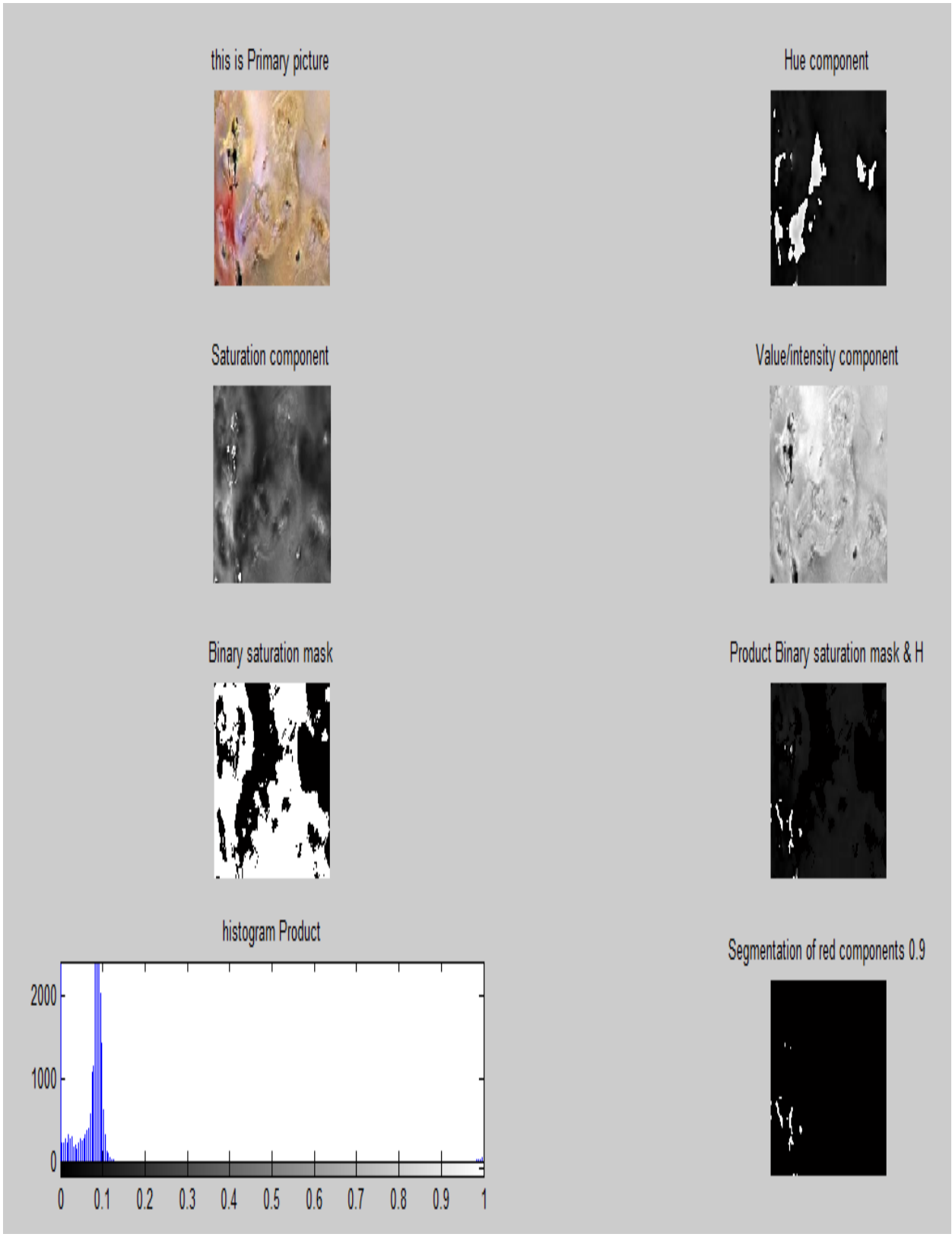
```
clc;
close all;
img=im2double(imread('C:\Users\Marjan\Documents\MATLAB\
30\1.jpg'));
%
img=imread('C:\Users\Marjan\Documents\MATLAB\30\Marjan.
jpg');
figure,subplot(4,2,1);
imshow(img);title('this is Primary picture');
% تـبـديـل از RGB بـه HSV
hsvTestImg = rgb2hsv(img);
% Segregate hue, sat و value/intensity
```

```
hueTestImg = hsvTestImg(:, :, 1);
satTestImg = hsvTestImg(:, :, 2);
valTestImg = hsvTestImg(:, :, 3);
                    RGB و اجزای HSV مربوطه را نمایش دهید
subplot(4,2, 2)
imshow(hueTestImg); title('Hue component');
subplot(4,2, 3)
imshow(satTestImg); title('Saturation component');
subplot(4,2, 4)
imshow(valTestImg); title('Value/intensity component');
% BW = imbinarize(satTestImg);
% threshold = graythresh(satTestImg);
BW=im2bw(satTestImg,0.29);0.29 باینری اشباع با آستانه
subplot(4,2, 5)
imshow(BW); title('Binary saturation mask');
product =immultiply(hueTestImg,BW); ضرب پرده رنگ در
دودویی اشباع
subplot(4,2, 6)
imshow(product); title('Product Binary saturation mask
& H');
subplot(4,2, 7)
imhist(product); title('histogram Product'); هیستوگرام
مرحله قبل
BW2=im2bw(product,0.9);
subplot(4,2, 8)
imshow(BW2); title('Segmentation of red components
0.9');0.9 باینری نتیجه با آستانه
```

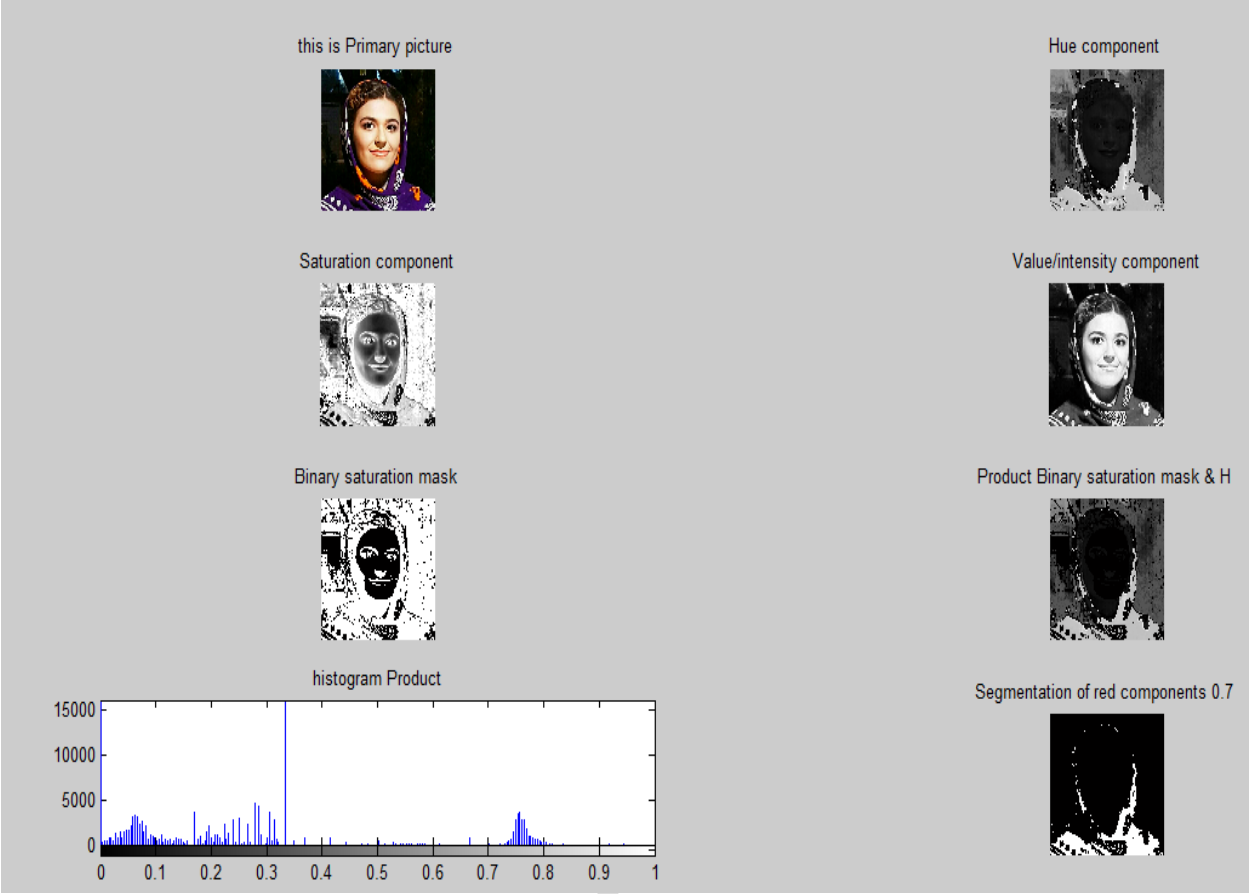
خروجی:

First I showed the hsi components, then I binary the saturation, with the next threshold, I multiplied the color screen in it and took its histogram, and finally, with a threshold of 0.9, I got an image whose white

dots are red in the main image.



And I divided the purple color of my scarf

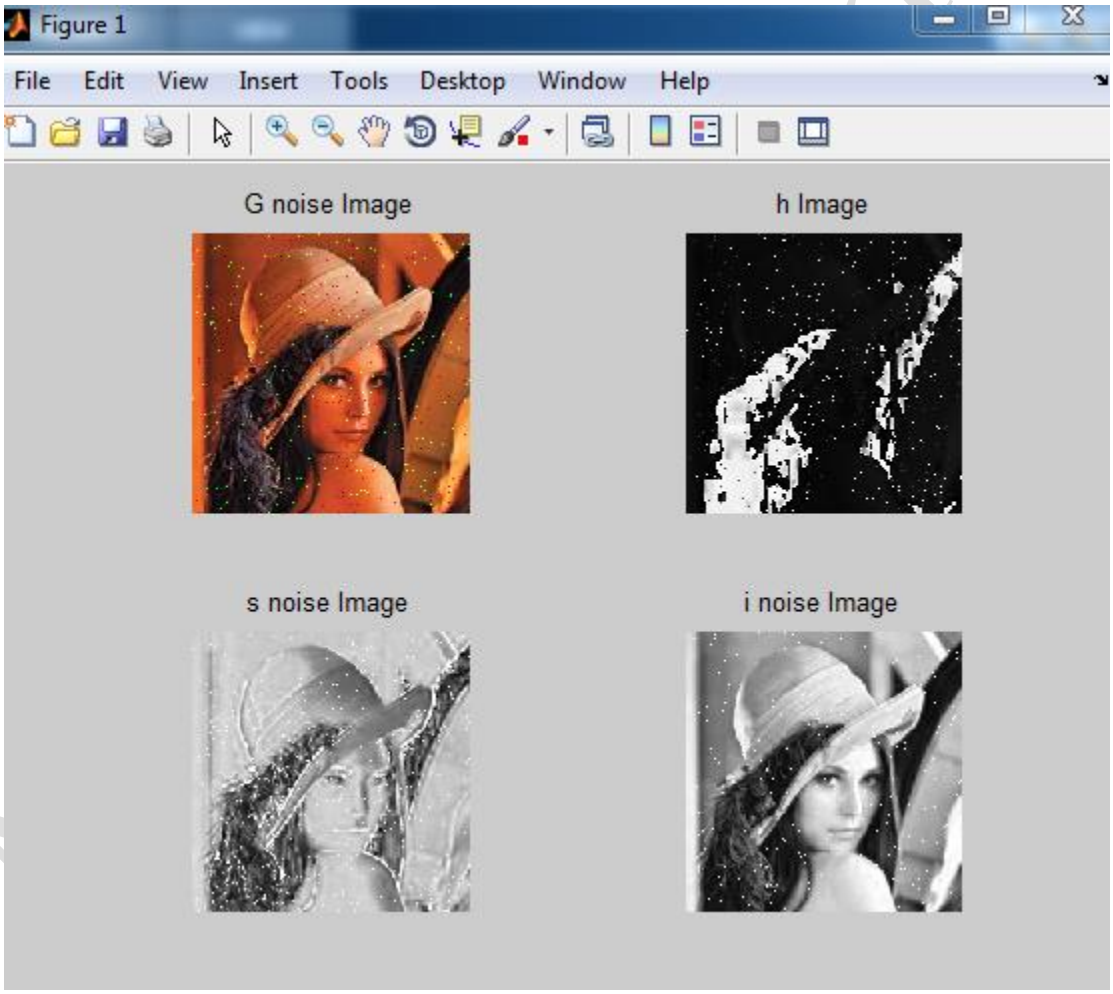


34) Take an rgb image and ruin the green channel with the noise of salt and pepper and bring it to hsi and display the components.

```
خواندن تصویر ورودی
input_image =
imread('C:\Users\Marjan\Documents\MATLAB\30\lena.jpg');
R = input_image(:,:,1); کانال های رنگ
G = input_image(:,:,2);
B = input_image(:,:,3);
J = imnoise(G,'salt & pepper',0.02); افزودن نویز نمک و
 فلفل به کانال سبز
input_image(:,:,1)=R;
input_image(:,:,2)=J;
input_image(:,:,3)=B;
hsvTestImg = rgb2hsv(input_image);      hsi تبدیل به
h = hsvTestImg(:,:,1);
s = hsvTestImg(:,:,2);
i = hsvTestImg(:,:,3);
```

```
نمایش پرده رنگ و اشباع و شدت
figure,subplot(2,2, 1)
imshow(input_image); title('G noise Image');
subplot(2,2, 2)
imshow(h); title('h Image');
subplot(2,2, 3)
imshow(s); title('s noise Image');
subplot(2,2, 4)
imshow(i); title('i noise Image');
```

Output: We only made green channel noise, but the noise was transmitted to all components of the hsi space.

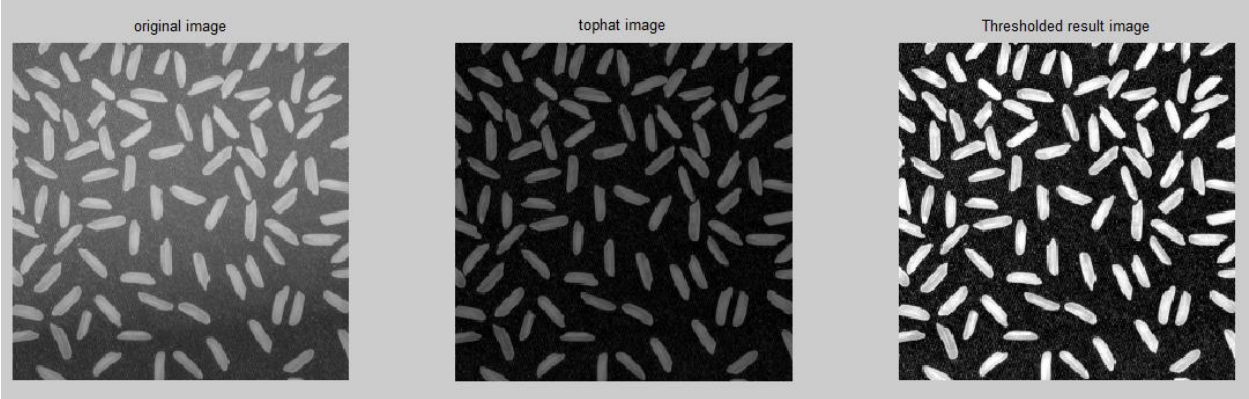


In cases where the noise affects only one rgb channel, the hsi converter distributes the noise to all hsi components.

35) Apply the top cap on the image.

```
Original = imread('rice.png');
figure,subplot(1,3,1)
imshow(original);title('original image');
se = strel('disk',12);  عنصر ساختاری
tophatFiltered = imtophat(original,se); تبدیل با لا کلاه
subplot(1,3,2)
imshow(tophatFiltered);title('tophat image');
contrastAdjusted = imadjust(tophatFiltered); آستانه
گذاری نتیجه
subplot(1,3,3)
imshow(contrastAdjusted);title('Thresholded result
image');
```

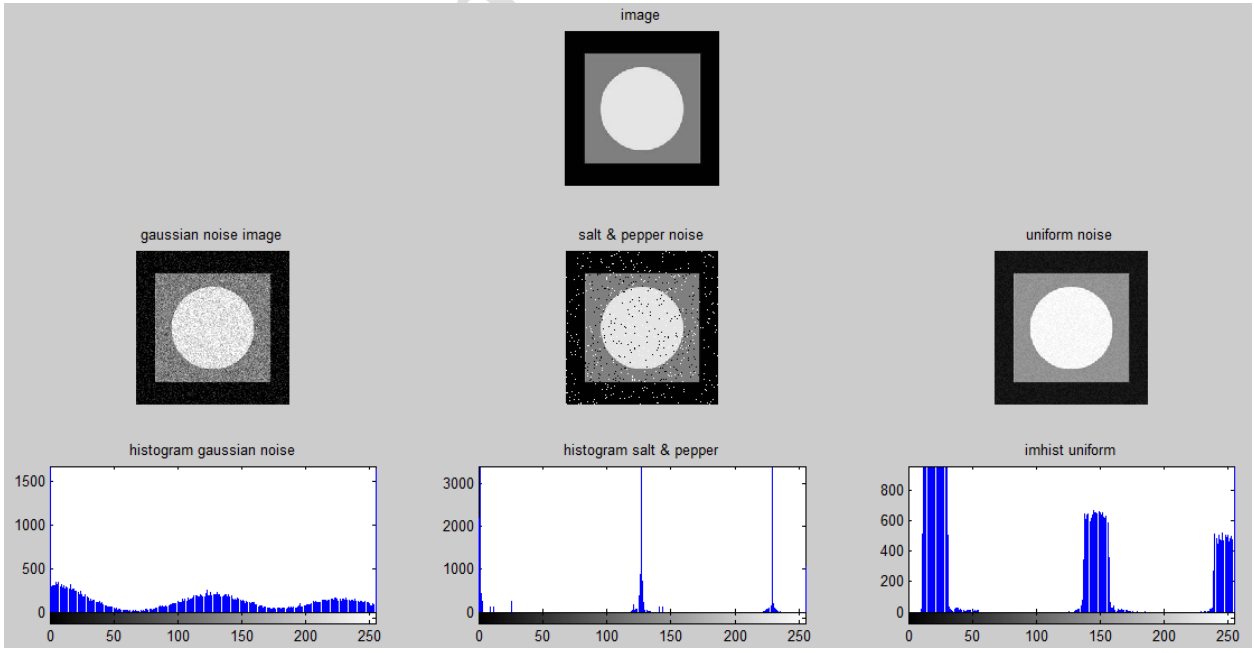
Output: Cap conversion for bright objects on a dark background.



36) Add Gaussian noise, salt, pepper and uniform to the image and draw the histogram of the image.

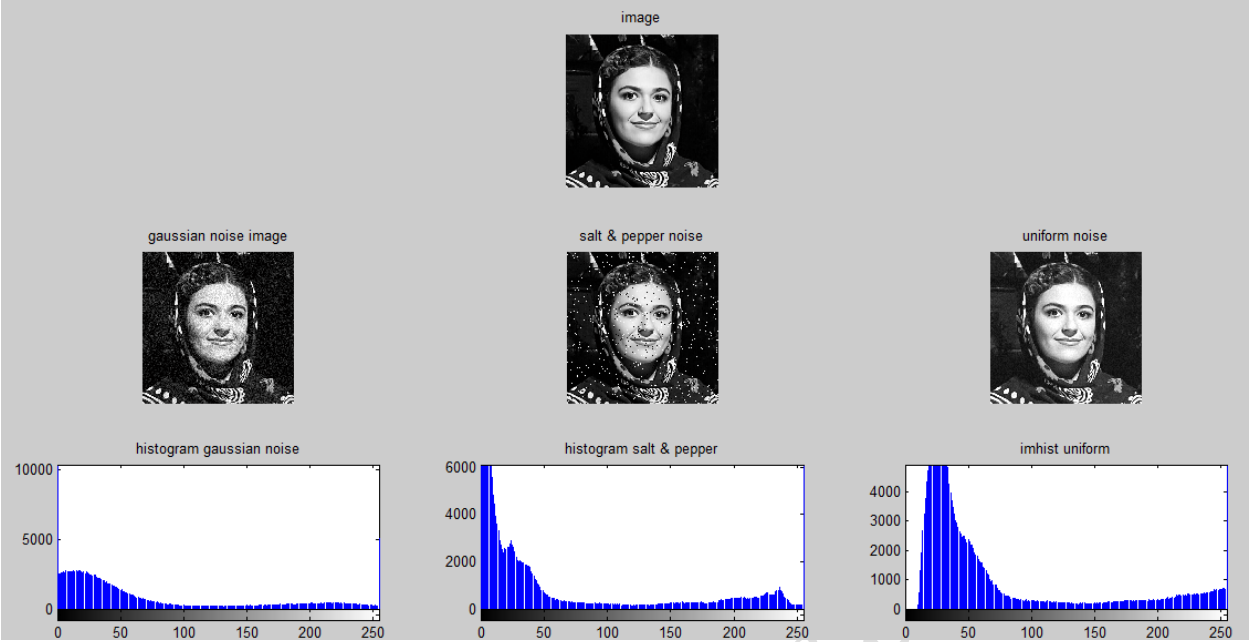
```
% I =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\Marjan.j
pg');
I =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\36\1.jpg
');
```

```
figure,subplot(3,3,2)
imshow(I);title('image');
I=rgb2gray(I);
J = imnoise(I,'gaussian'); نویز گوسی
subplot(3,3,4)
imshow(J);title('gaussian noise image');
subplot(3,3,7)
imhist(J); title('histogram gaussian noise');هیستوگرام
J2 = imnoise(I,'salt & pepper');
subplot(3,3,5)
imshow(J2);title('salt & pepper noise');  نویز نمک و
فلفل
subplot(3,3,8)
imhist(J2);title('histogram salt & pepper');هیستوگرامش
A = 10;
B = 30;
matrix_uniform = A + (B-A)*rand(size(I)); نویز یکنواخت
noisy_image = uint8(I) + uint8(matrix_uniform);
subplot(3,3,6)
imshow(noisy_image);title('uniform noise');
subplot(3,3,9)
imhist(noisy_image);title('imhist uniform');
```



The noise image histogram is the shape of the noise.

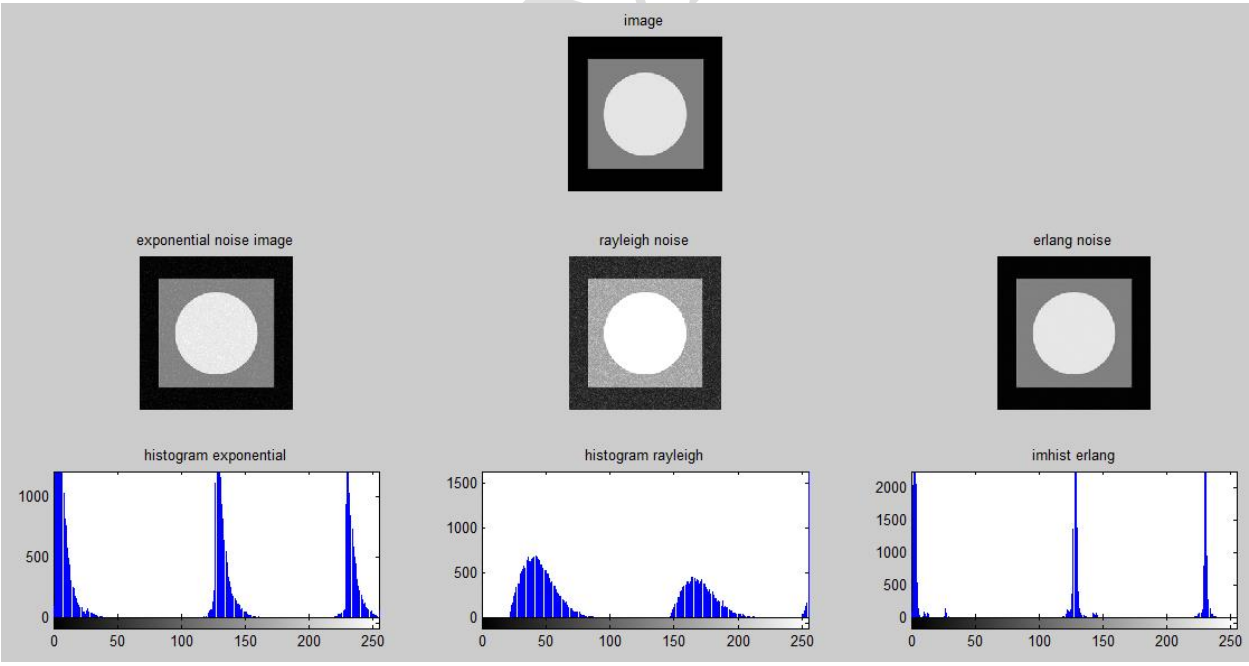
June 20, 2022



37) Add Erlang, Riley and power noise to the image and draw the image histogram.

```
% I =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\Marjan.j
pg');
I =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\36\1.jpg
');
figure,subplot(3,3,2)
imshow(I);title('image');
I=rgb2gray(I);
a=0.2;
b=0.1;
[M,N]=size(I);
if a<=0
   error('parameter a must be positive for
exponential');
end
k=-1/a;
R=k*log(1-rand(M,N)); نویز توانی
J=uint8(R)+uint8(I); افزودن به تصویر
subplot(3,3,4)
imshow(J);title('exponential noise image');
subplot(3,3,7)
imhist(J); title('histogram exponential');
```

```
aa=20;
bb=700;
J2=aa+(-bb*log(1-rand(M,N))).^.5; نویز رایلی
JJ=uint8(J2)+uint8(I); افزودن به تصویر
subplot(3,3,5)
imshow(JJ);title('rayleigh noise');
subplot(3,3,8)
imhist(JJ);title('histogram rayleigh');
ae=2;be=3;
k=-1/ae;
R=zeros(M,N);
for j=1:be
R=R+k*log(1-rand(M,N)); نویز ارلانگ
end
d=uint8(R)+uint8(I); افزودن به تصویر
subplot(3,3,6)
imshow(d);title('erlang noise');
subplot(3,3,9)
imhist(d);title('imhist erlang');
```

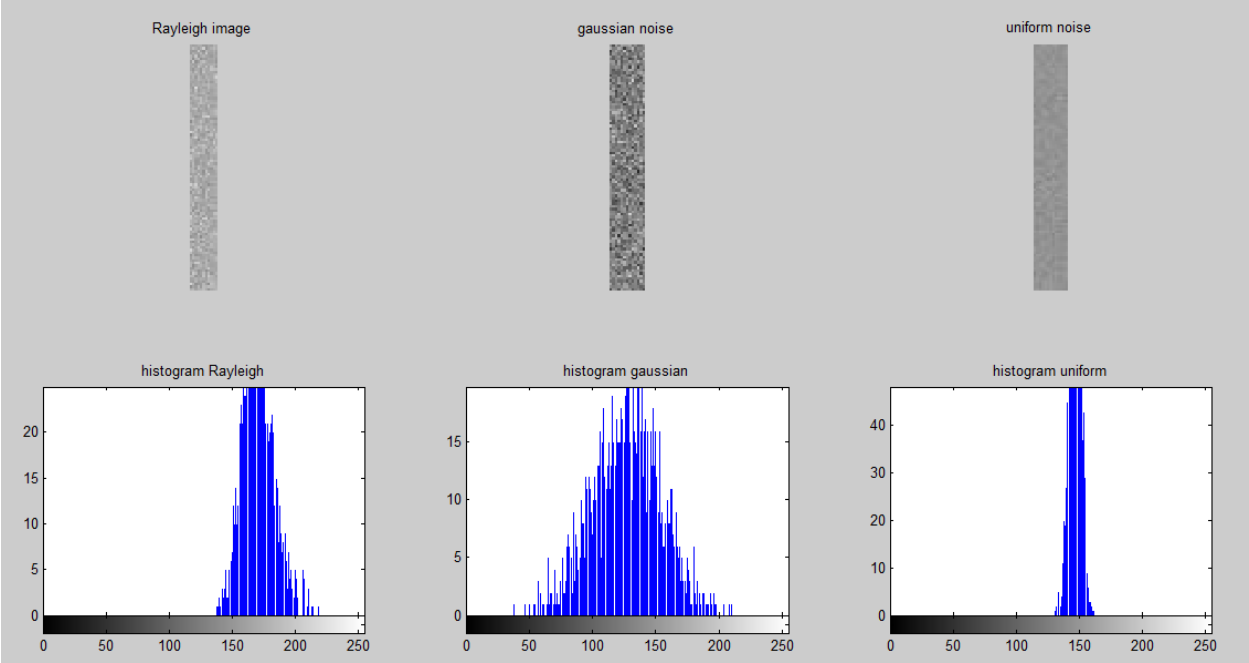


The noise image histogram is the shape of the noise.

38) Estimate the noise parameter based on a part of the image with a uniform gray color.

June 20, 2022

```
% I =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\Marjan.j
pg');
I =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\36\37\38
\Rayleigh.jpg');
figure,subplot(2,3,1)
imshow(I);title('Rayleigh image');
I=rgb2gray(I);
subplot(2,3,4)
imhist(I);title('histogram Rayleigh');  هیستگرام
I2 =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\36\37\38
\Gaussian.jpg');
subplot(2,3,2)
imshow(I2);title('gaussian noise');
I2=rgb2gray(I2);
subplot(2,3,5)
imhist(I2);title('histogram gaussian'); هیستگرام
I3 =
imread('C:\Users\Marjan\Documents\MATLAB\34\35\36\37\38
\uniform.jpg');
subplot(2,3,3)
imshow(I3);title('uniform noise');
I3=rgb2gray(I3);
subplot(2,3,6)
imhist(I3);title('histogram uniform'); هیستگرام
```



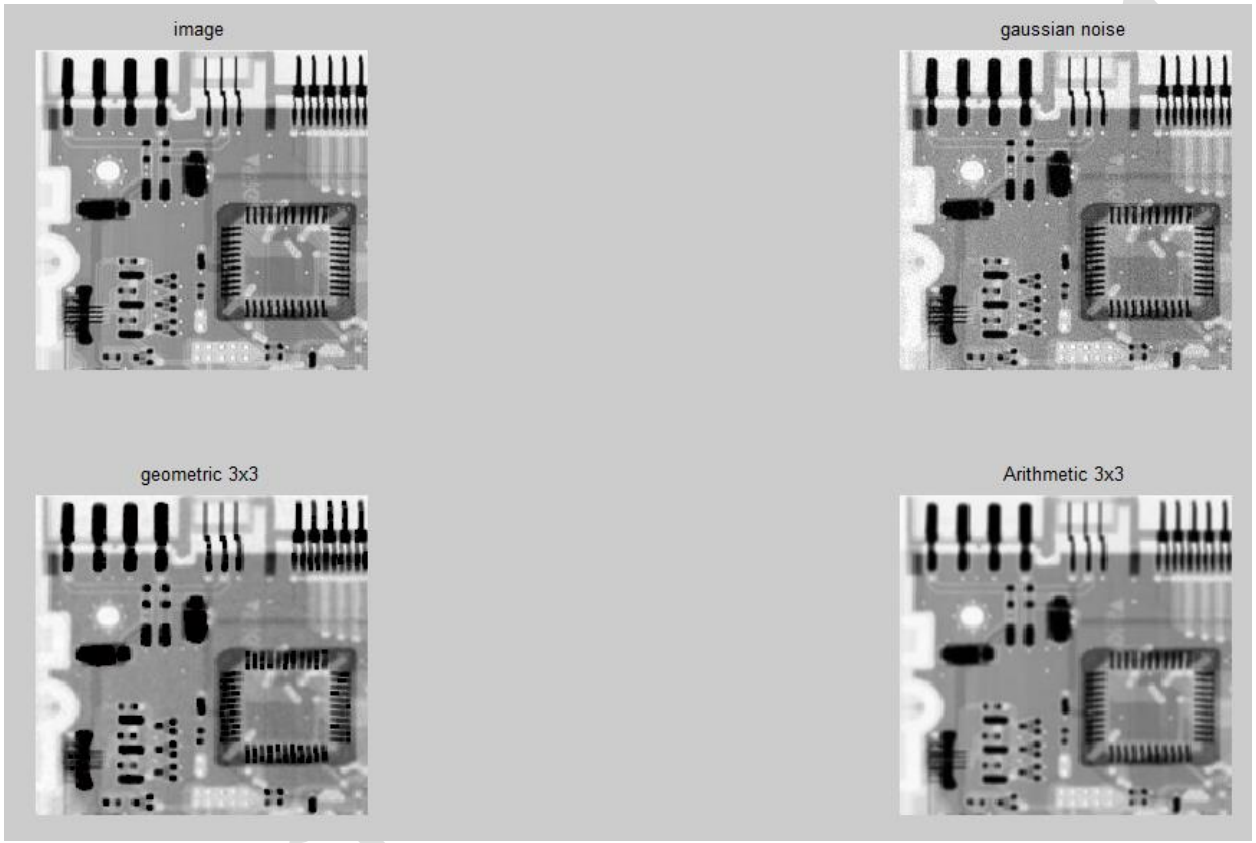
The histogram of these noisy gray images is quite similar to the noise applied to the whole image.

39) Take an image and apply Gaussian noise to it and then filter it with the arithmetic mean filter.

```
% I =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\Marjan.j
pg');
I =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\3.jpg');
figure,subplot(2,2,1)
I=rgb2gray(I);
imshow(I);title('image');
J = imnoise(I,'gaussian',0,0.001); نویز گوسی به تصویر
subplot(2,2,2)
imshow(J);title('gaussian noise');
g=im2double(J);
kr=3;
kc=3;
f=exp(imfilter(log(g),ones(kr,kc),'replicate')).^(1/(kr
*kc)); فیلتر هندسی
subplot(2,2,3)
```

```
imshow(f);title('geometric 3x3');
w=ones(kr,kc)/(kr*kc);
F=imfilter(I,w,'replicate','same');فیلتر میانگین حسابی
subplot(2,2,4)
imshow(F);title('Arithmetic 3x3');
```

The geometric filter does not blur and is sharper than the arithmetic filter.

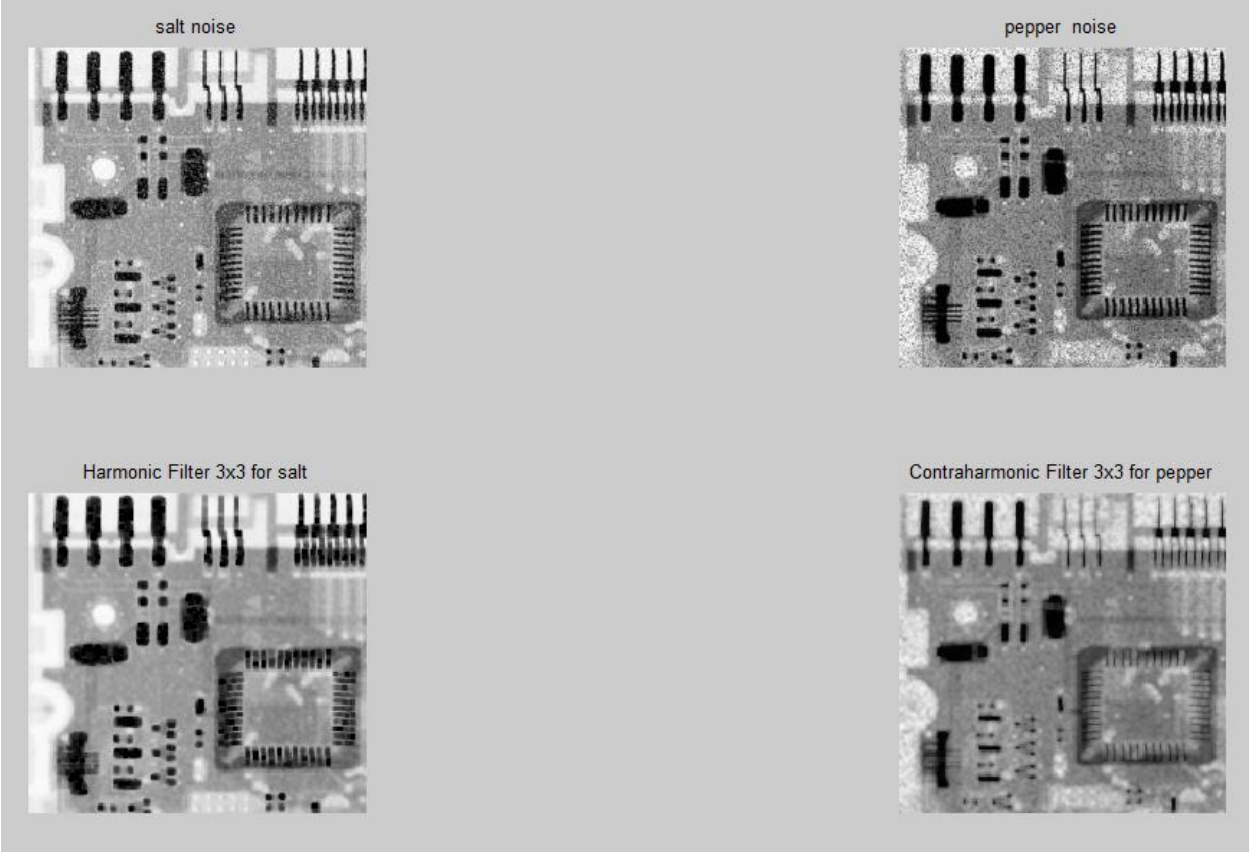


40) Apply a harmonic filter on an image with salt noise and an anti-harmonic filter on an image with pepper noise.

```
% I =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\Marjan.jpg');
I =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\40\33.jpg');
I=rgb2gray(I);تصویر با نویز نمک خاکستری کن
```

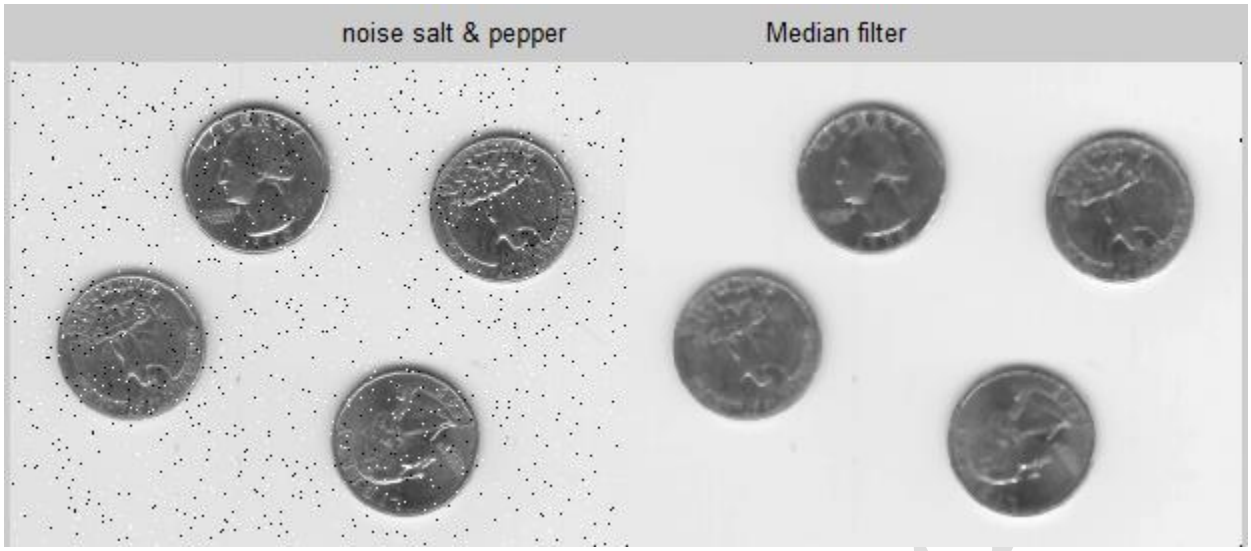
```
I2 =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\40\34.jp
g');
I2=rgb2gray(I2);تصویر با نویز فلفل خاکستری کن
si=I;
gi=I2;
figure,subplot(2,2,1)
imshow(si);title('salt noise');
subplot(2,2,2)
imshow(gi);title('pepper  noise');
sg=im2double(si);
gg=im2double(gi);
ord=1;
kr=3;
kc=3;
sf=(kr*kc)./imfilter(1./(sg+eps),ones(kr,kc),'replicate
'); فیلتر هارمونیک
gf=imfilter(gg.^(ord+1),ones(kr,kc),'replicate')./(imfi
lter(gg.^(ord),ones(kr,kc),'replicate')+eps); فیلتر ضد
 هارمونیک
subplot(2,2,3)
imshow(sf);title('Harmonic Filter 3x3 for salt');
subplot(2,2,4)
imshow(gf);title('Contraharmonic Filter 3x3 for
pepper');
```

For my own image, I first applied salt and pepper noise to an image. I eliminated the salt noise with a harmonic.



41) Apply a moderate filter of salt and pepper noise.

```
% I =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\Marjan.j
pg');
% I=rgb2gray(I);
I = imread('eight.tif');
figure, imshow(I)
J = imnoise(I,'salt & pepper',0.02); نویز نمک و فلفل
K = medfilt2(J); فیلتر میانه
imshowpair(J,K,'montage'), نمایش دو نیجه کنارهم
title('noise salt & pepper
Median filter');
```

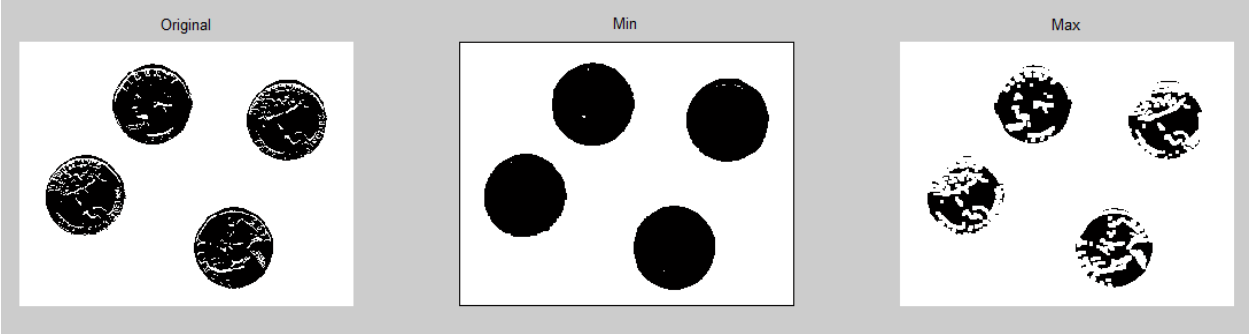


answered very well.

42) Apply min and max to an image.

```
% Original =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\Marjan.j
pg');
% Original=rgb2gray(Original);
Original= imread('eight.tif');

BW = im2bw(Original,0.5);                تبدیل به سیاه و
سفید

minf=@(x) min(x(:));                     'min()' تنظیم فیلتر
maxf=@(x)max(x(:));                      'max()' تنظیم فیلتر
min_Image=nlfilter(BW,[3 3],minf);   3 x 3 همسایگی
max_Image=nlfilter(BW,[3 3],maxf);   3 x 3
figure,
subplot(1,3,1), imshow(BW), title('Original');
%Display imageنتیجه
subplot(1,3,2), imshow(min_Image), title('Min');
%Display min image
subplot(1,3,3), imshow(max_Image), title('Max');
%Display max image
```

In min the image is dark and max the image is light.



43) Midpoint filter

```
Original =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\Marjan.j
pg');
Original=rgb2gray(Original);
% Original= imread('eight.tif');

BW = im2bw(Original,0.5);                تبدیل به سیاه و
سفید

minf=@(x) min(x(:));                     'min()' تنظیم فیلتر
maxf=@(x)max(x(:));                      'max()' تنظیم فیلتر
min_Image=nlfilter(BW,[3 3],minf);  3 x 3 همسایگی
max_Image=nlfilter(BW,[3 3],maxf);  3 x 3 همسایگی
d=(min_Image+max_Image)/2; فیلتر نقطه میانی
figure,
subplot(1,2,1), imshow(BW), title('Original');
%Display image
subplot(1,2,2), imshow(d), title('Midpoint filter');
```

We take the minimum and the maximum and divide by 2.

Gives a result between minimum and maximum.

44) Adaptive local noise reduction filter

```
% A = imread('saturn.png');
A =
imread('C:\Users\Marjan\Documents\MATLAB\34\39\Marjan.j
pg');
B = rgb2gray(A);
sz = size(B,1)*size(B,2);


افزودن نویز گوسی با میانگین 0 و واریانس 0.005
B = imnoise(B,'gaussian',0,0.005);
figure,imshow(B); title('Image with gaussian noise');

 B = double(B);

سایز پنجره
M = 5;
N = 5;

ماتریس صفر در همه طرف قرار دهید
C = padarray(B,[floor(M/2),floor(N/2)]);


lvar = zeros([size(B,1) size(B,2)]);
lmean = zeros([size(B,1) size(B,2)]);
```

```
temp = zeros([size(B,1) size(B,2)]);
NewImg = zeros([size(B,1) size(B,2)]);

for i = 1:size(C,1)-(M-1)
    for j = 1:size(C,2)-(N-1)


        temp = C(i:i+(M-1),j:j+(N-1));
        tmp =  temp(:);
              % میانگین محلی و واریانس محلی برای منطقه
محلی را بیابید
        lmean(i,j) = mean(tmp);
        lvar(i,j) = mean(tmp.^2)-mean(tmp).^2;

    end
end

درصد واریانس نویز و میانگین واریانس محلی
nvar = sum(lvar(:))/sz;

%اگر noise_variance > local_variance ،
local_variance=noise_variance
 lvar = max(lvar,nvar);

 %Final_Image = B- (نویز واریانس/واریانس محلی)*(B-
local_mean);
 NewImg = nvar./lvar;
 NewImg = NewImg.*(B-lmean);
 NewImg = B-NewImg;

%تبدیل تصویر به فرمت uint8.
 NewImg = uint8(NewImg);
figure,imshow(NewImg);title('Restored Image using
Adaptive Local filter');
```

Adaptive filter is for when we do not do the same for all images, use the middle for intensity and variance for contrast. He answered very well.

45) Adaptive middle filter.

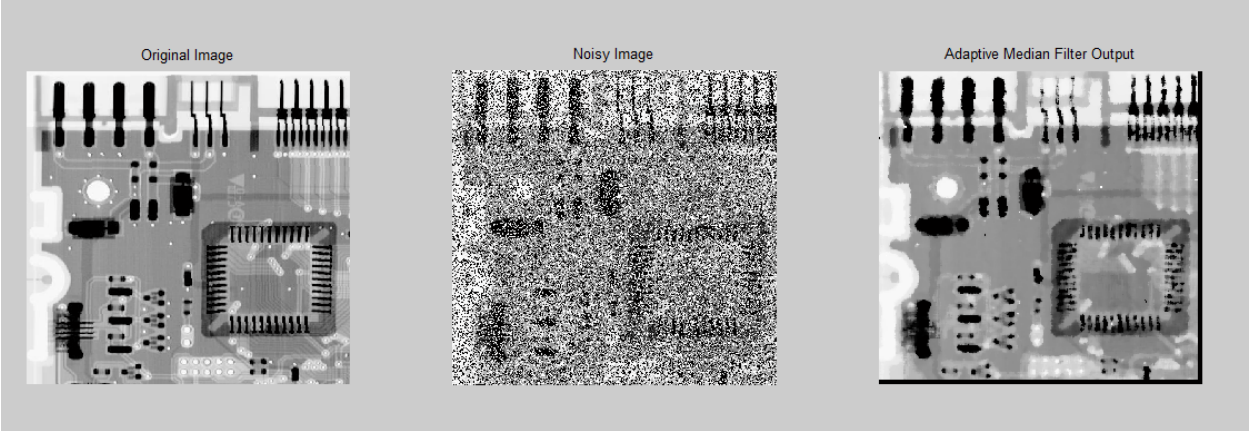
```
function adapmedfilter(testimage,window)
    paddedimage=padarray(testimage,[1,1]);
%Padding the array
    [m,n]= size(paddedimage);
اندازه ماتریس پر شده %
    Wmax=7;
حداکثر اندازه پنجره
    output=zeros(m,n);
% خروجی تصویر

    for i=1:m-(window-1)
        for j=1:n-(window-1)
            C=paddedimage(i:i+(window-1),j:j+(window-
1));                    %پنجره
            D=C(:)';
% تبدیل از 2 بعدی به 1 بعدی
            [u,v]=size(D);
```

```
            for p=1:v-1
% الگوریتم مرتب سازی
                for q=1:v-1
                    if D(q) > D(q+1)
                        temp= D(q+1);
                        D(q+1)=D(q);
                        D(q)=temp;
                    end
                end
            end

            zmin=D(u);
% حداقل مقدار پیکسل در پنجره
            zmax=D(v);
پیکسل با حداکثر مقدار در پنجره
            k=round((window*window)/2);
            zxy=D(k);
%Pixel در (x,y)
            p=round((u+v)/2);
            if mod(v,2) == 0
% محاسبه میانه
                zmed=D(p+1);

            else
                zmed=D(p);

            end

            A1= zmed-zmin;
% الگوریتم اصلی شروع می شود
            A2= zmed-zmax;
            if A1>0 && A2<0
                B1=zxy-zmin;
                B2=zxy-zmax;
                if B1>0 && B2<0
                    output(i,j)=zxy;
               else
                    output(i,j)=zmed;
                end
            else
                windowsize=window+1;
```

```
                if  windowsize <= Wmax

adapmedfilter(testimage,windowsize);
% فراخوانی تابع
                    return
                else
                    output(i,j)=zmed;
                end

            end
%پایان الگوریتم اصلی

        end
    end
    oadap=uint8(output); %تبدیل تصویر خروجی به فرمت
uint8
    figure(1) % نمایش نتیجه
    subplot(1,3,1); imshow('I.tif'); title('Original
Image');
    subplot(1,3,2); imshow('Marjan_imagenoisy.tif');
title('Noisy Image');
    subplot(1,3,3);imshow(oadap); title('Adaptive
Median Filter Output');
end

%
image=rgb2gray(imread('C:\Users\Marjan\Documents\MATLAB
\34\39\Marjan.jpg'));
clc; clear all; close all;
testimage=imread('C:\Users\Marjan\Documents\MATLAB\41\A
daptive-Median-Filter-master\Adaptive-Median-Filter-
master\Marjan_imagenoisy.tif');
%تصویر تست
W=input('Enter the size of the window');
%اندازه پنجره
adapmedfilter(testimage,W);
% فراخوانی تابع
```

I entered the window size 6:

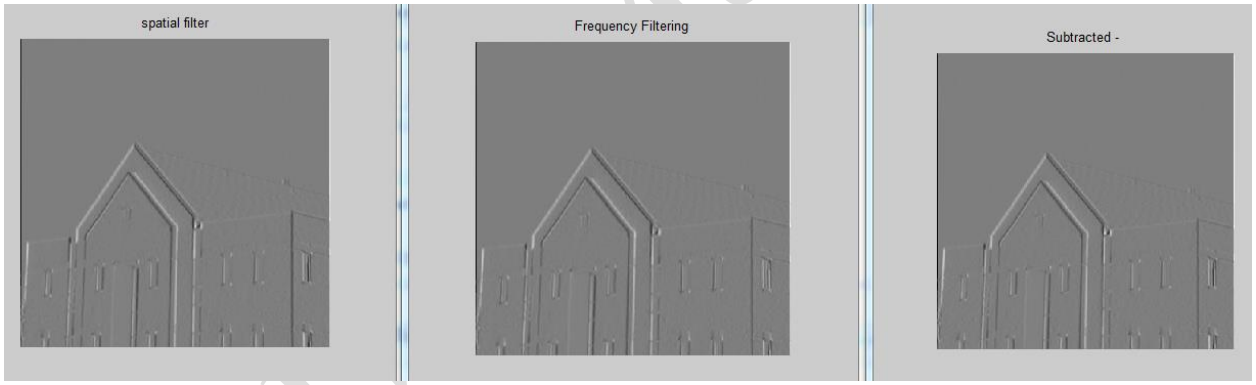
(Suitable for when the probability of impact noise density above 0.2)



46) Take an image and apply the Sobel vertical filter in the frequency and location domain on it and get the difference.

```
% تصویر فیلتر مکانی را ایجاد کنید
f =
im2double(rgb2gray(imread('C:\Users\Marjan\Documents\MA
TLAB\34\39\Marjan.jpg')));
% f =
im2double(rgb2gray(imread('C:\Users\Marjan\Documents\MA
TLAB\41\Adaptive-Median-Filter-master\Adaptive-Median-
Filter-master\46\1.jpg')));
h = [-1 0 1;-2 0 2; -1 0 1]; فیلترعمودی سوبل
sfi = imfilter(f,h, 'conv'); اعمال فیلتر
% نمایش نتایج (نمایش همه مقادیر)
figure,imshow(sfi, []); title('spatial filter');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
تصویر فیلتر شده با فرکانس را ایجاد کنید
f =
im2double(rgb2gray(imread('C:\Users\Marjan\Documents\MA
TLAB\34\39\Marjan.jpg')));
% f =
rgb2gray(imread('C:\Users\Marjan\Documents\MATLAB\41\Ad
aptive-Median-Filter-master\Adaptive-Median-Filter-
master\46\1.jpg'));
h = [-1 0 1;-2 0 2; -1 0 1];فیلترعمودی سوبل
[m,n] = size(f);
```

```
% 2 ایجاد یک آرایه تهی به اندازه m X 2n
c = zeros(2*m,2*n);
% خواندن اندازه آرایه تهی
PQ = size(c);
F = fft2(double(f), PQ(1), PQ(2));تبدیل فوریه تصویر
H = fft2(double(h), PQ(1), PQ(2));تبدیل فوریه فیلتر
F_fH = H.*F; ضرب تبدیل های فوریه درهم
ffi = ifft2(F_fH); عکس تبدیل فوریه
ffi = ffi(2:size(f,1)+1, 2:size(f,2)+1);حذف صفرها

% نمایش نتایج (نمایش همه مقادیر)
figure, imshow(ffi,[]);title('Frequency Filtering');
 تفاضل نتیجه مکانی و فرکانسی
dblSubtractedImage = double(ffi) - double(sfi);
figure,imshow(dblSubtractedImage, []);title('Subtracted
- ');
As we can see, despite the similarity of the results of
.the two areas, they are different
```



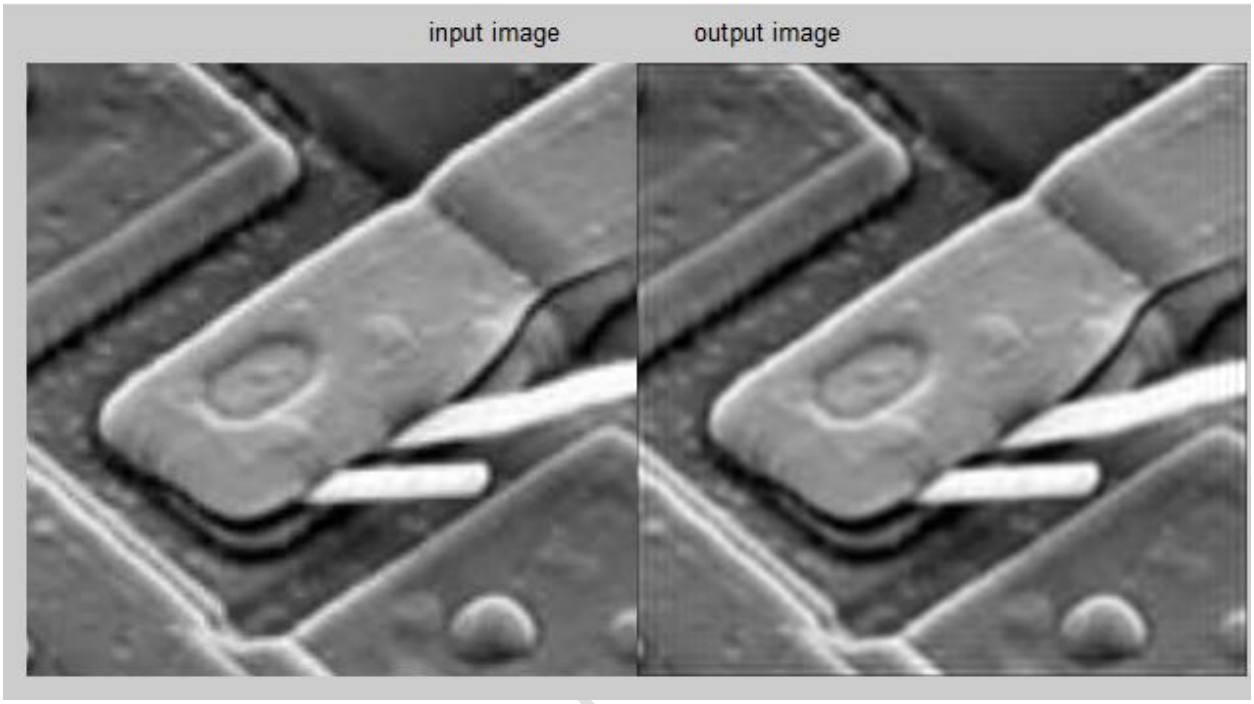
47) Frequency filtering steps. (Low-pass filter)

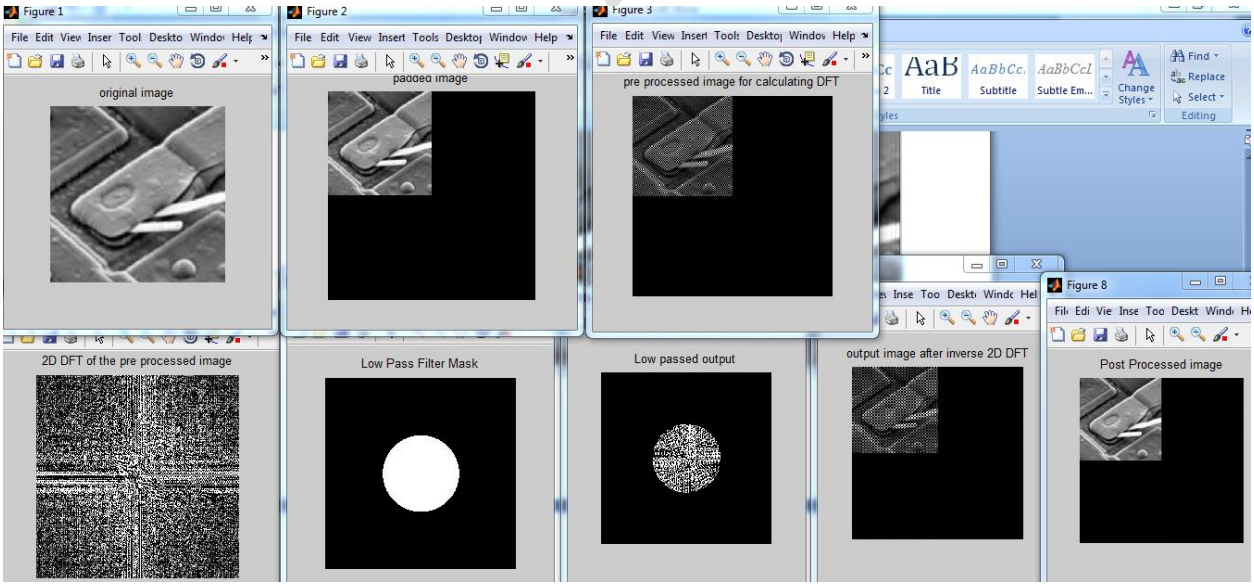
```
clc;
clear all;
close all;
a =
rgb2gray(imread('C:\Users\Marjan\Documents\MATLAB\50\1.
jpg'));
b = im2double(a);
```

```
 [m,n] = size(b);
c = zeros(2*m,2*n);
 [p,q] = size(c);
```
اضافه کردن تصویر اصلی به آرایه صفر برای ایجاد یک پدینگ %
```
for i = 1:p
    for j = 1:q
        if i <= m && j<= n
            c(i,j) = b(i,j);
        else
            c(i,j) = 0;
        end
    end
end
imshow(b);title('original image');
figure;
imshow(c);title('padded image');
% 3 مرحله
% ایجاد یک آرایه 0 به اندازه p X q
d = zeros(p,q);
                          (-1)^(x+y)%ضرب تصویر پر شده با
for i = 1:p
    for j = 1:q
        d(i,j) = c(i,j).*(-1).^(i + j);
    end
end
figure;
imshow(d);title('pre processed image for calculating
DFT');
% 4 مرحله
          "fft2"%محاسبه DFT دو بعدی با استفاده از دستور
e = fft2(d);
figure;imshow(e);title('2D DFT of the pre processed
image');
%%%%%%%%%%%%%%%%%%%
% 5 مرحله
   از  را با استفاده Pass Filter"   "Low %در اینجا ما یک
                  % پیاده سازی می کنیم matlab "freqspace"
[x,y] = freqspace(p,'meshgrid');
z = zeros(p,q);
for i = 1:p
```

```
    for j = 1:q
        z(i,j) = sqrt(x(i,j).^2 + y(i,j).^2);
    end
end
% انتخاب فرکانس قطع و در نتیجه تعریف فیلتر پایین گذر
% ماسک
H = zeros(p,q);
for i = 1:p
    for j = 1:q
        if z(i,j) <= 0.4  فرکانس قطع 0.4 در اینجا % LPF
است
            H(i,j) = 1;
        else
            H(i,j) = 0;
        end
    end
end
figure;imshow(H);title('Low Pass Filter Mask');
h1 = e.*H;
figure;
imshow(h1);title('Low passed output');
       : gp(x,y) = {real{inverse DFT[G(u,v)]}(-7 مرحله%
                                                1)^(x+y)
                              "out"محاسبه DFT 2 بعدی معکوس%

h2 = ifft2(h1);
figure;
imshow(h2);title('output image after inverse 2D DFT');
% عملیات پس از فرآیند
h3 = zeros(p,q);
for i = 1:p
    for j = 1:q
        h3(i,j) = h2(i,j).*((-1).^(i+j));
    end
end
figure;
imshow(h3);title('Post Processed image');
out = zeros(m,n);
for i = 1:m
    for j = 1:n
        out(i,j) = h3(i,j);
```

```
    end
end
figure;
imshow([b out]);title('input image
output image');
```





We added it to filter 0, scaled it, took Fourier transform from it, prepared a filter in the frequency space, which can be a function in space when we take Fourier transform and multiply it and take the photo of Fourier transform. We see that the bottom filter has blurred the image.

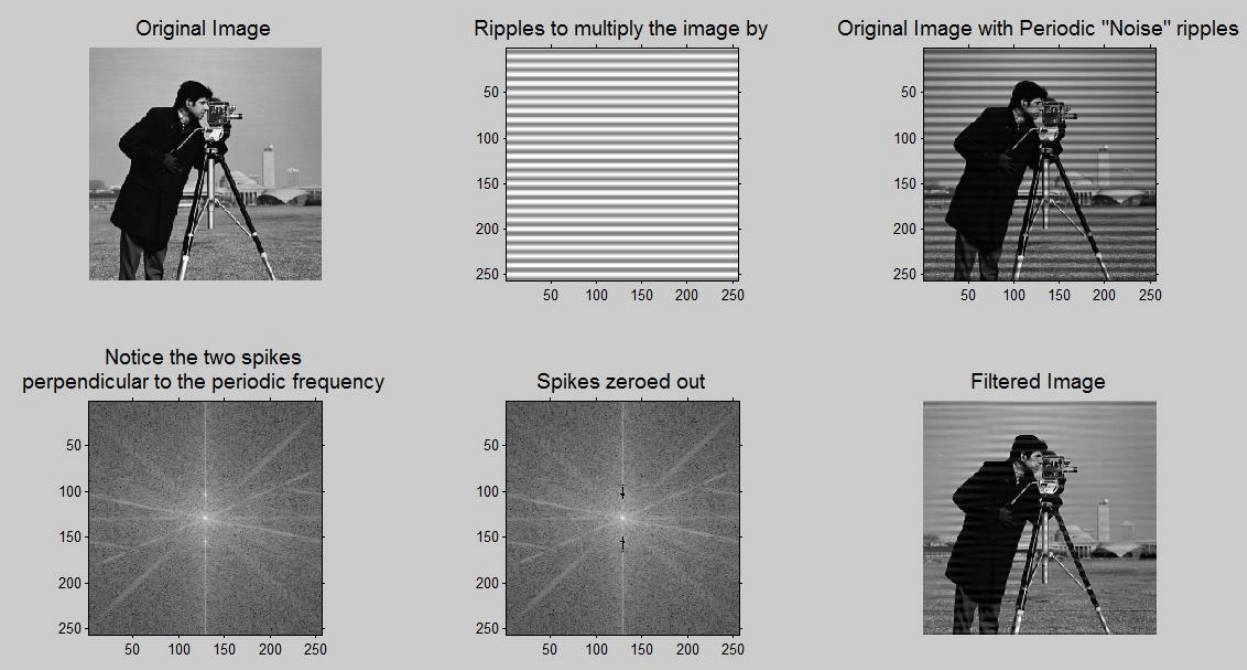
48) Periodic noise pattern removal (with phage filter)

```
% فیلتر کردن یک تصویر، با ریپل دوره ای، در دامنه فوریه.
clc;
close all;
clear;
workspace;
format long g;
format compact;
fontSize = 14;

% در یک تصویر آزمایشی استاندارد متلب در مقیاس خاکستری بخوانید.
folder = fileparts(which('cameraman.tif')); % تعیین کنید پوشه دمو کجاست (با همه نسخه ها کار می کند).
baseFileName = 'cameraman.tif';
% نام فایل کامل را با مسیر از پیش تعیین شده دریافت کنید.
fullFileName = fullfile(folder, baseFileName);
% بررسی کنید که آیا فایل وجود دارد
if ~exist(fullFileName, 'file')
    % فایل وجود ندارد -- آن را در آنجا پیدا نکردم. مسیر جستجوی آن را بررسی کنید.
    fullFileName = baseFileName; % این بار مسیری وجود ندارد.
    if ~exist(fullFileName, 'file')
% هنوز آن را پیدا نکردم. کاربر هشدار
        errorMessage = sprintf('Error: %s does not exist in the search path folders.', fullFileName);
        uiwait(warndlg(errorMessage));
        return;
    end
end
```

```
grayImage = imread(fullFileName);
[rows, columns ,numberOfColorBands] = size(grayImage);
if numberOfColorBands > 1
    grayImage = rgb2gray(grayImage);
end
subplot(2, 3, 1);
imshow(grayImage, [0 255]);
set(gcf, 'Name', ['Results for ' fullFileName]);
title('Original Image', 'FontSize', fontSize);
set(gcf, 'units','normalized','outerposition',[0 0 1
1]); % به حداکثر رساندن رقم.

% موج های بزرگی به آن اضافه کنید.
rowVector = (1 : rows)';
period = 10; % 20 rows
amplitude = 0.5; اندازه ریپل %.
offset = 1 - amplitude; چقدر کسینوس بالاتر از 0 است %.
cosVector = amplitude * (1 + cos(2 * pi * rowVector /
period))/2 + offset;
ripplesImage = repmat(cosVector, [1, columns]);
subplot(2, 3, 2);
minValue = min(min(ripplesImage));
maxValue = max(max(ripplesImage));
imshow(ripplesImage, [0 maxValue]);
axis on;
title('Ripples to multiply the image by', 'FontSize',
fontSize);
 %ریپل ها را در تصویر ضرب کنید تا تصویری با "نویز" دوره
                                        ای در آن به دست آید.
grayImage = ripplesImage .* double(grayImage);
minValue = min(min(grayImage));
maxValue = max(max(grayImage));
subplot(2, 3, 3);
imshow(grayImage, [0 255]);
axis on;
title('Original Image with Periodic "Noise" ripples',
'FontSize', fontSize);

                                    بعدی را محاسبه کنید.% fft 2
frequencyImage = fftshift(fft2(grayImage));
```

```
% بزرگی log را بگیرید تا بتوانیم آن را بهتر در صفحه نمایش ببینیم.
amplitudeImage = log(abs(frequencyImage));
minValue = min(min(amplitudeImage));
maxValue = max(max(amplitudeImage));
subplot(2, 3, 4);
imshow(amplitudeImage, []);
caption = sprintf('Notice the two spikes\nperpendicular
to the periodic frequency');
title(caption, 'FontSize', fontSize);
axis on;
% zoom(10)

% محل خوشه های بزرگ را پیدا کنید.
amplitudeThreshold = 10.9;
brightSpikes = amplitudeImage > amplitudeThreshold; %
تصویر باینری
subplot(2, 3, 5);
imshow(brightSpikes);
axis on;
title('Bright Spikes', 'FontSize', fontSize);
% اجازه دهید کاربر تصویر را ببیند.
promptMessage = sprintf('The image below shows the
bright spikes.\nNow we will exclude the central
spike.');
titleBarCaption = 'Continue?';
button = questdlg(promptMessage, titleBarCaption,
'Continue', 'Cancel', 'Continue');
if strcmpi(button, 'Cancel')
    return;
end
% خوشه DC مرکزی را حذف کنید. همه چیز از ردیف 115 تا
143.

brightSpikes(115:143, :) = 0;
imshow(brightSpikes);
title('Bright spikes other than central spike',
'FontSize', fontSize);

promptMessage = sprintf('Now we will use these bright
spikes to filter (mask) the spectrum.');
```

```
button = questdlg(promptMessage, titleBarCaption,
'Continue', 'Cancel', 'Continue');
if strcmpi(button, 'Cancel')
    return;
end
طیف را فیلتر/ماسک کنید %.
frequencyImage(brightSpikes) = 0;
% بزرگی log  را بگیرید تا بتوانیم آن را بهتر در صفحه
نمایش ببینیم.
amplitudeImage2 = log(abs(frequencyImage));
minValue = min(min(amplitudeImage2));
maxValue = max(max(amplitudeImage2));
subplot(2, 3, 5);
imshow(amplitudeImage2, [minValue maxValue]);
axis on;
title('Spikes zeroed out', 'FontSize', fontSize);
% zoom(10)

filteredImage = ifft2(fftshift(frequencyImage));
amplitudeImage3 = abs(filteredImage);
minValue = min(min(amplitudeImage3));
maxValue = max(max(amplitudeImage3));
subplot(2, 3, 6);
imshow(amplitudeImage3, [minValue maxValue]);
title('Filtered Image', 'FontSize', fontSize);
% set(gcf, 'units','normalized','outerposition',[0 0 1
1]); % Maximize figure.
% %
```



Periodic noise pattern with phage filter is very low.

49) Location-independent linear degradation (blur image as a result of motion) and Weiner filtering to retrieve the image.

```
I =
rgb2gray(imread('C:\Users\Marjan\Documents\MATLAB\49\1.
jpg'));
% I =
rgb2gray(imread('C:\Users\Marjan\Documents\MATLAB\49\Ma
rjan.jpg'));
figure; subplot(1,3,1),imshow(I); title('Original
Image');
PSF = fspecial('motion',21,11);
Idouble = im2double(I);
blurred = imfilter(Idouble,PSF,'conv','circular');
subplot(1,3,2), imshow(blurred); title('Blurred
Image');
wnr1 = deconvwnr(blurred,PSF);
subplot(1,3,3),imshow(wnr1)
title('Restored Blurred Image(Wiener) Filtering')
```

To illustrate, this example takes a clear image and intentionally blurs it by tangling it with a PSF filter. The example uses the fspecial function to create a PSF filter that simulates a motion blur, specifying the blur length in pixels (LEN = 31) and the blur angle in degrees (THETA = 11). Once the PSF is created, the instance uses the imfilter function to blend the PSF with the original image, I, to create a blurred image. Recover the blurry image using the deconvwnr function to find out how much the reverse process is. The image has no blur, so we can delete the Noise-to-Signal (NSR) input argument.

50) Simulation and recovery of Motion Blur and Gaussian Noise This time there is also noise.

```
Ioriginal = imread('cameraman.tif');
figure; subplot(2,4,1),imshow(Ioriginal)
title('Original Image')
PSF = fspecial('motion',21,11);
Idouble = im2double(Ioriginal);
blurred = imfilter(Idouble,PSF,'conv','circular');
subplot(2,4,2),imshow(blurred)
title('Blurred Image')
%new
noise_mean = 0;
noise_var = 0.0001;
blurred_noisy =
imnoise(blurred,'gaussian',noise_mean,noise_var);
subplot(2,4,3),imshow(blurred_noisy)
title('Blurred and Noisy Image')
wnr2 = deconvwnr(blurred_noisy,PSF);
subplot(2,4,4),imshow(wnr2)
title('Restoration of Blurred Noisy Image (NSR = 0)')
signal_var = var(Idouble(:));
NSR = noise_var / signal_var;
```

```
wnr3 = deconvwnr(blurred_noisy,PSF,NSR);
subplot(2,4,5),imshow(wnr3)
title('Restoration of Blurred Noisy Image (Estimated
NSR)')
```

Using the imnoise function, add zero medium Gaussian noise to the blurred image. Try to retrieve the blurred image using deconvwnr without providing noise estimation. By default, the Wiener NSR recovery filter assumes 0. In this case, the Wiener recovery filter is equivalent to an ideal reverse filter that can be extremely sensitive to noise in the input image. In this example, the noise in the recovery is amplified to the point that the image content is lost. Using deconvwnr, we retrieve the blurred image with a more realistic estimated amount of noise.