OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

EE417POST-LABREPORT-1

Scaling and Filtering in Grayscale Images

OSMAN BURAK ÇEVİK

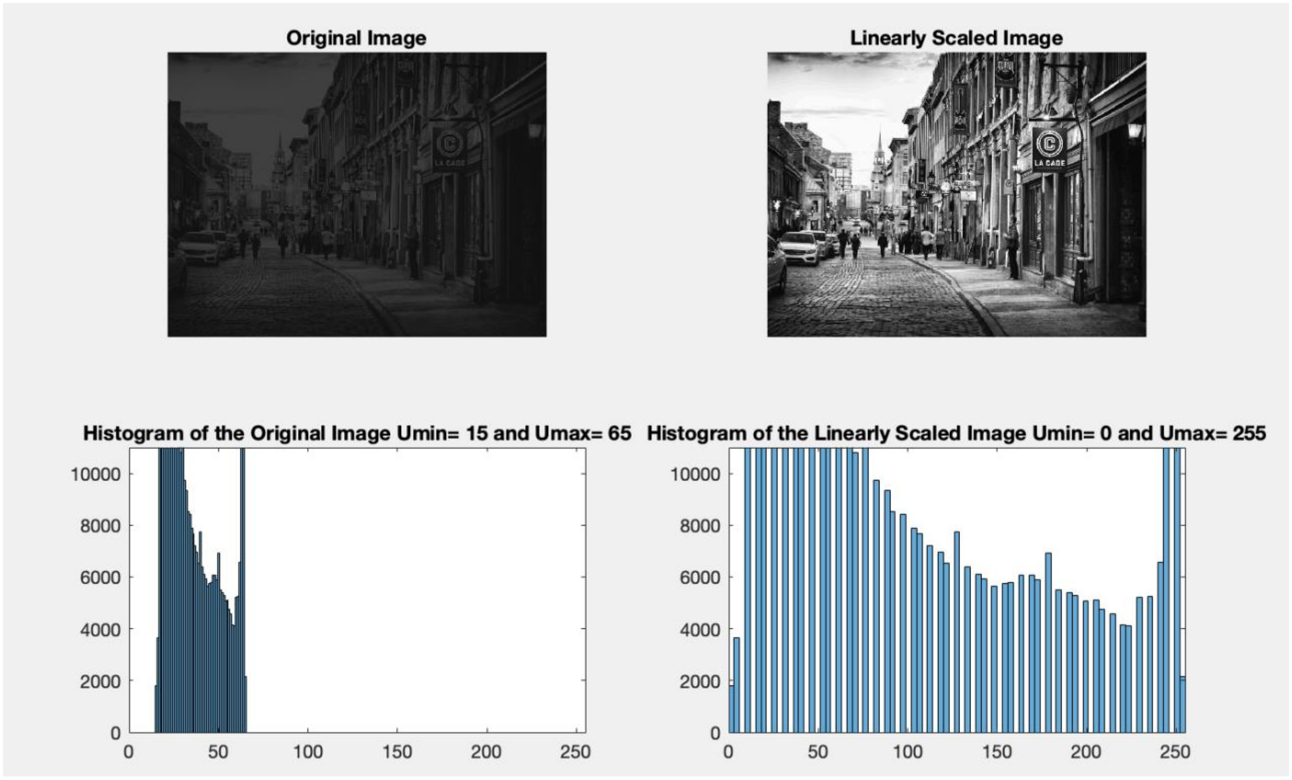
26399 | osmanburak@sabanciuniv.edu

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

FUNCTION-1: lab1linscale.m

The goal of the function lab1linscale.m is to linearly spread the values used in the image(Im) to the scale 0-Gmax(Assuming the image Im has positive histogram values in an interval).

Results for lab1linscale.m function when the image is "city.png":



OBSERVATIONS:

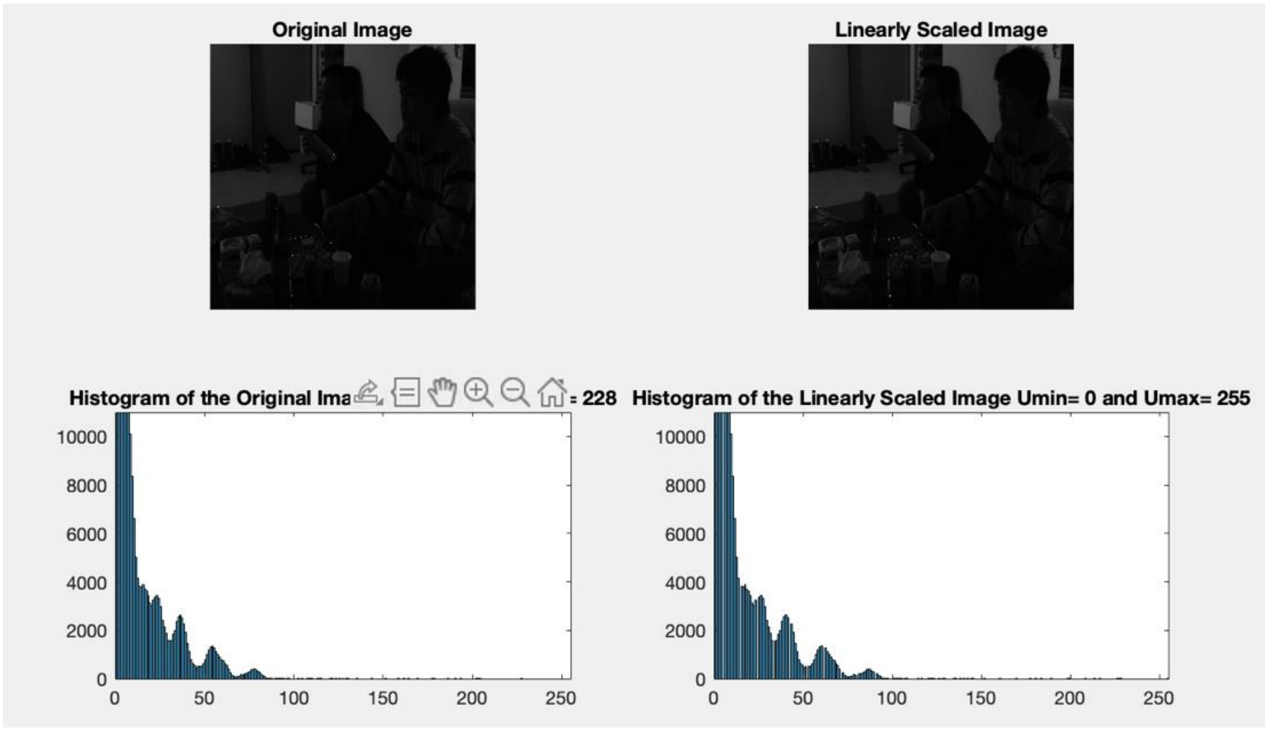
The scale of original image is between 0-65.5, however after the process of lab1linscale.m function the new scale of the proccesed image is 0-255.

The goal of the function lab1linscale.m is to linearly spread the values of the image onto the scale 0-Gmax. In the code Gmax=255 because of uint8() conversion.

"city.png" is 768x576 and function works properly.

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

Results for lab1linscale.m function when the image is "darkImage.png":



OBSERVATIONS:

The scale of original image is between 0-90, however after the process of lab1linscale.m function the new scale of the proccesed image is 0-255.

The goal of the function lab1linscale.m is to linearly spread the values of the image onto the scale 0-Gmax. In the code Gmax=255 because of uint8() conversion.

"darkImage.png" is 1054x794 and function works properly.

FUNCTION-1 CONCLUSIONS:

Lab1linscale.m function linearly spreads the values of the image to the scale 0-Gmax and the function works properly for different sized images. Window size is not a paramaterer for the function.

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

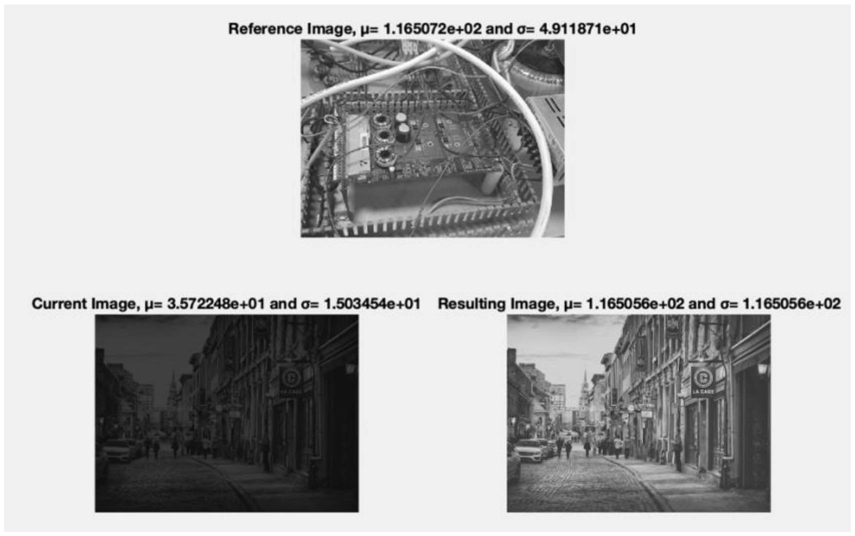
FUNCTION-1: lab1condscale.m

Lab1condscale.m takes image Im as the image to be processed and Iref as the reference image that will scale the image Im as parameters. The function returns Im, the reference image Iref and the scaled image Jnew.

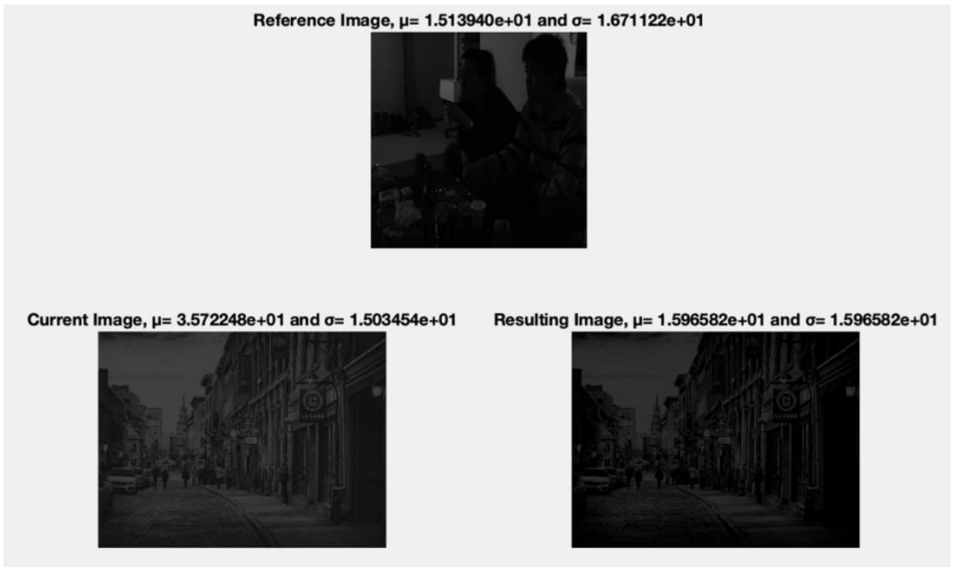
The function is a point operator function(The function changes the value of pixels in image Im with the mean and varience of the reference image Iref).

Image Im is mapped with the mean and varience of the reference image Iref and stored as Jnew.

Results for lab1condscale.m function when the image is “city.png” and the reference image Iref is “board.jpg”:


Reference Image, μ= 1.165072e+02 and σ= 4.911871e+01
Current Image, μ= 3.572248e+01 and σ= 1.503454e+01
Resulting Image, μ= 1.165056e+02 and σ= 1.165056e+02

Results for lab1condscale.m function when the image is “city.png” and the reference image Iref is “darkImage.png”:


Reference Image, μ= 1.513940e+01 and σ= 1.671122e+01
Current Image, μ= 3.572248e+01 and σ= 1.503454e+01
Resulting Image, μ= 1.596582e+01 and σ= 1.596582e+01

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

OBSERVATIONS:

By only changing the reference images I've tried to understand and observe the impact of different reference images Irefs to the process.

Mean pixel values of "board.jpg" are closer values to 255. Therefore the new image Jnew has greater pixel values.

Mean pixel values of "darkImage.png" are closer values to 0. Therefore the new image Jnew has lower pixel values.

Size the refernce images are; city.png: 768x576 and darkImage.png: 510x510.

FUNCTION-2 CONCLUSIONS:

Lab1condscale.m function linearly spreads the values of image Im to the scale of the reference image Iref.

Image width & height is not a parameter for the function this it works properly for different sized images.

If the reference image has greater mean of pixel values than the current image the resulting image has greater mean pixel values as well, if the reference image has lower mean of pixel values than the current image the resulting image has lower mean pixel values as well.

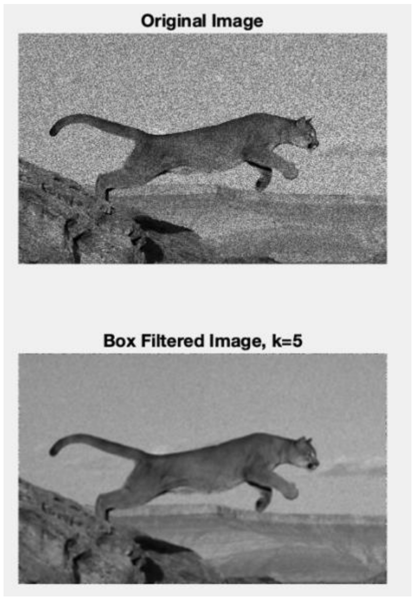
OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

FUNCTION-3: lab1locbox.m

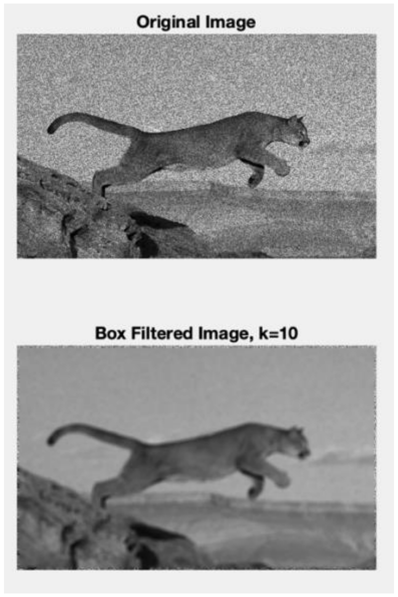
Lab1locbox.m function takes an image Im and an integer k as inputs and returns the image Im and the filtered version of the image Im as Inew.

Box filter(local mean filter) is a local operator used for reducing noise in the target image Im. Operator convolves the image Im with a (2k+1)x(2k+1) sliding window Wp. Box filter replaces each pixel within an image with the average of its neighboorhood.

Results for lab1locbox.m function when the image is "jump.png" and k=5:

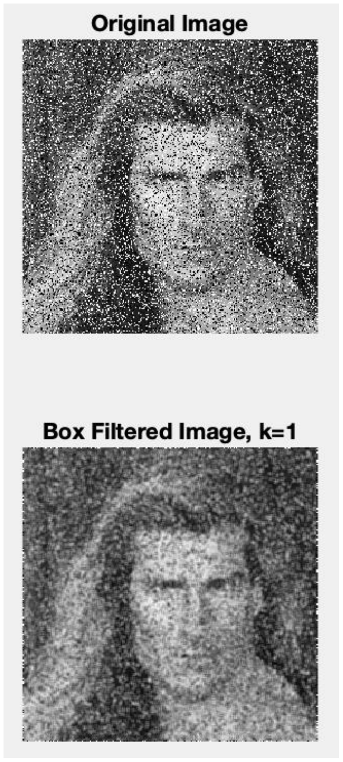


Results for lab1locbox.m function when the image is "jump.png" and k=5:

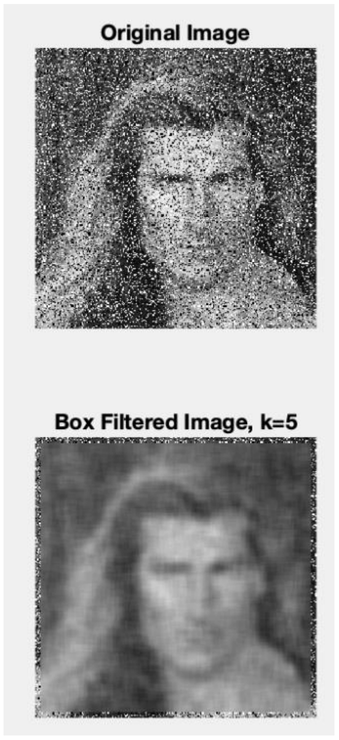


OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

Results for lab1locbox.m function when the image is “aaaa.png” and k=1:



Results for lab1locbox.m function when the image is “aaaa.png” and k=5:



OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

OBSERVATIONS:

By only changing the parameter k I've tried to understand and observe the impact of the size of the sliding window Wp. When k increases, the sliding window also increases. With greater sized Wp, salt & pepper(noise) of the filtered image Im decreases. With greater sized Wp, the filtered image Im gets blurred.

Size of jump.png: 1577x986.
Size of aaaa.png: 256x256

FUNCTION-3 CONCLUSIONS:

Lab1locbpx.m function linearly creates a sliding window with a size of (2k+1)x(2k+1). The input k decides the size of the sliding window. As Wp increases to a breaking point the original image's nois decreases and gets blurred. After the breaking point the blurred pixel amount decreases significantly and unflitered pixels occur at the edges.

The function works properly for different sized images therefore the size of the image does not affect the function.

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

FUNCTION-4: lab1maxmin.m

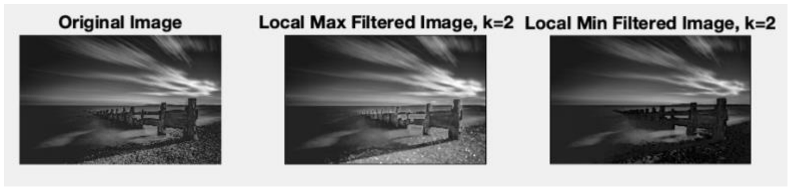
Lab1locmaxmin.m function takes an image Im and an integer k as inputs and returns Im and the filtered versions of the image, Imin and Imax.

The function contains 2 local operators: local maximum and local minimum. Local maximum and local minimum operators are used for dilation and erosion of pixels in images. Operators convolve with a (2k+1)x(2k+1) sized sliding window Wp.

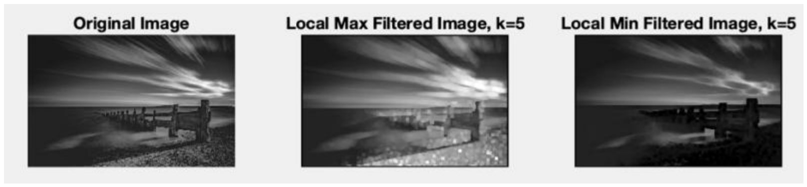
Local maximum filter replaces each pixel within an image with the maximum of its neighboorhood.

Local minimum filter replaces each pixel within an image with the minimum of its neighboorhood.

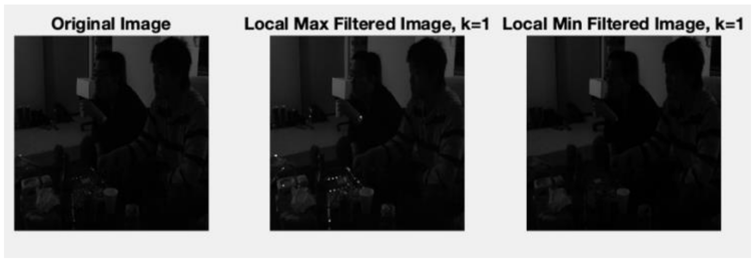
Results for lab1locmaxmin.m function when the image is “currentImage.png” and k=2:



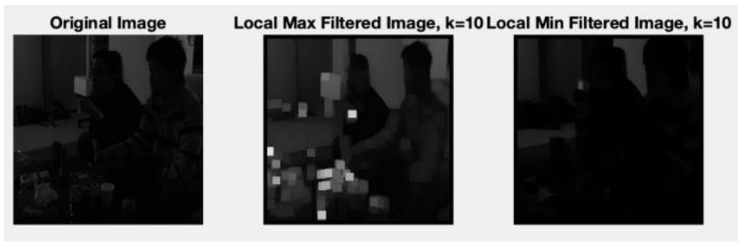
Results for lab1locmaxmin.m function when the image is “currentImage.png” and k=5:



Results for lab1locmaxmin.m function when the image is “darkImage.png” and k=1:



Results for lab1locmaxmin.m function when the image is “darkImage.png” and k=10:


Original Image
Local Max Filtered Image, k=10
Local Min Filtered Image, k=10

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

OBSERVATIONS:

By increasing the value of the input k, I've tried to understand and observe the affect of the size of the sliding window Wp.

The values of the pixels increase with the local maximum filter. 255 is the highest pixel value for uint8().

The values of the pixels decrease with the local minimum filter. 0 is the lowest pixel value for uint8().

Size of currentImage.png: 728x464
Size of darkImage.png: 510x510

FUNCTION-4 CONCLUSIONS:

Lab1locmaxmin.m function linearly creates a sliding window with a size of (2k+1)x(2k+1).

As the size of the sliding window Wp increases, local maximum filter updates the original image's each pixel with the maximum of its neighboorhood. Therefore the filtered image gets greater pixel values.(The average of pixels increases).

As the size of the sliding window Wp increases, local minimum filter updates the original image's each pixel with the minimum of its neighboorhood. Therefore the filtered image gets lower pixel values.(The average of pixels decreases).

Image size is not a parameter for the function.

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

My histogram equilization algorithm:

```
function myHistEq()

current = imread('darkImage.png');

[r,c,~] = size(current);

new = uint8(zeros(r,c));

res = zeros(256,1);

cum = zeros(256,1);

cdf = zeros(256,1);

pdf = zeros(256,1);

f = zeros(256,1);

num = r*c;

for i = 1:1:r

    for j = 1:1:c

        x = current(i,j);
        f(x+1) = f(x+1)+1;
        pdf(x+1) = f(x+1)/num;

    end

end

sum =0 ;

val = 255;

for i = 1:1:size(pdf)

    sum=f(i)+sum;

    cum(i)=sum;

    cdf(i)=cum(i)/ num;

    res(i) =round(val*cdf(i));

end
```

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

```
for i = 1:1:r

    for j = 1:1:c

        new(i,j) = res(current(i,j)+1);

    end

end
```

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

APPENDIX

```
function [Inew]=lab1linscale(Im)
[h,w,c]=size(Im);
if c==3
    Im=rgb2gray(Im);
end

I=double(Im);
a= -min(I(:));
Gmax=255;
b= Gmax/(max(I(:))-min(I(:)));
Inew=b*(I+a);
Inew=uint8(Inew);

subplot(2,2,1),imshow(Im);
title 'Original Image'

subplot(2,2,2),imshow(Inew);
title 'Linearly Scaled Image'

subplot(2,2,3),histogram(Im);
xlim([0,255])
ylim([0,11000])
formatSpec="Histogram of the Original Image Umin= %d and Umax= %d";
A1=(min(I(:)));
A2=(max(I(:)));
str=sprintf(formatSpec,A1,A2)
title(str)

subplot(2,2,4),histogram(Inew);
xlim([0,255])
ylim([0,11000])
formatSpec2="Histogram of the Linearly Scaled Image Umin= %d and Umax= %d";
A3=(min(Inew(:)));
A4=(max(Inew(:)));
str2=sprintf(formatSpec2,A3,A4)
title(str2)
end
```

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

```
function [Jnew]=lab1condscale(Im,Iref)
[~,~,c]=size(Im);
if c==3
    Im=rgb2gray(Im);
end

[~,~,c]=size(Iref);
if c==3
    Iref=rgb2gray(Iref);
end

J=double(Im);
I=double(Iref);

b=std(I(:))/std(J(:));
a=mean(I(:))/b-mean(J(:));
Jnew=b*(J+a);

disp([mean(I(:)),mean(J(:)),mean(Jnew(:))]);
disp([std(I(:)),std(J(:)),std(Jnew(:))]);

Jnew=uint8(Jnew);

subplot(2,2,1.5),imshow(Iref);
formatSpec="Reference Image, Œ°= %d and œÉ= %d";
A1=(mean(I(:)));
A2=(std(I(:)));
str=sprintf(formatSpec,A1,A2)
title(str)

subplot(2,2,3),imshow(Im);
formatSpec2="Current Image, Œ°= %d and œÉ= %d";
A3=(mean(J(:)));
A4=(std(J(:)));
str2=sprintf(formatSpec2,A3,A4)
title(str2)

subplot(2,2,4),imshow(Jnew);
formatSpec3="Resulting Image, Œ°= %d and œÉ= %d";
A5=(mean(Jnew(:)));
A6=(mean(Jnew(:)));
str3=sprintf(formatSpec3,A5,A6)
title(str3)
end
```

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

```
function [Inew]=lab1locbox(Im,k)
I=double(Im);
[h,w,c]=size(I);
Inew=I;

for i=k+1:h-k
    for j=k+1:w-k
        wp=I(i-k:i+k,j-k:j+k);
        Inew(i,j)=mean(wp(:));
    end
end

Inew=uint8(Inew);

subplot(2,1,1),imshow(Im);
title 'Original Image'

subplot(2,1,2),imshow(Inew);
formatSpec="Box Filtered Image, k=%d";
A1=(k);
str=sprintf(formatSpec,A1)
title(str)
end
```

OSMAN BURAK ÇEVİK
osmanburak@sabanciuniv.edu
26399

```
function [Imax,Imin]=lab1locmaxmin(Im,k)
I=double(Im);
[h,w,c]=size(I);
Imax=zeros(h,w);
Imin=zeros(h,w);

for i=k+1:h-k
    for j=k+1:w-k
        wp=I(i-k:i+k ,j-k:j+k);
        Imax(i,j)=max(wp(:));
        Imin(i,j)=min(wp(:));
    end
end

Imax=uint8(Imax);
Imin=uint8(Imin);

subplot(1,3,1),imshow(Im);
title 'Original Image'

subplot(1,3,2),imshow(Imax);
formatSpec="Local Max Filtered Image, k=%d";
A1=(k);
str=sprintf(formatSpec,A1)
title(str)

subplot(1,3,3),imshow(Imin);
formatSpec2="Local Min Filtered Image, k=%d";
str2=sprintf(formatSpec2,A1)
title(str2)
end
```