Group 02

# Project 5-Local Image Enhancement


## Group 02

Mehrnaz Tavan(8709203189)- Robin Kanthe (8105194917)


### Table of Contents

# 1-Introduction

In this project, we try to apply local enhancement methods to an image which has some dark hidden parts. The methods applied in this study consist of local histogram enhancement and enhancement using local histogram statistical properties.

Part two describes the problem at hand in more details. In part three, the applied methods are discussed. The results will come in part four. The MATLAB codes used for this project will appear in part five.

# 2-Problem Description

**"5. Local image enhancement**

The main reason for applying image enhancement is to make the image proper for a special purpose. Histogram equalization is a type of spatial domain image enhancement which is used to make the details more visible. In this method, the gray levels of the image are transformed using a certain transformation function, *T(r)*. Histogram equalization is performed in the following way:

Initially, the normalized histogram of the image is computed. To create the transformation function, the cumulative sum of the normalized histogram is calculated and rounded to the nearest gray level. This cumulative sum serves as the transformation function between the input gray levels and the output gray levels. It can be formulated mathematically as

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \quad (1)$$

where is the output gray level, the input gray level, the transformation function and the probability of occurrence of an input gray level .

The result is an image which has a more variety of gray levels and consequently higher contrast. Increasing the contrast in an image in most cases increases the amount of visible details. This is usable in many applications, for example when examining X-ray images, trying to find a small fracture in a bone.

One disadvantage with this method is that it can enhance noise in images since it increases the contrast between non-noisy parts and noisy parts. This is particularly visible in images with large, uniform areas. Another disadvantage becomes visible when dealing with images which have large dark areas and also some very bright areas. The method equalizes the histogram so that the dark pixels are moved closer to the light ones. The result will be an image which appears too bright and somewhat "washed out". Another problem is that sometimes different parts of the image have very different characteristics. One part of the image could contain a lot of details with gray levels very similar to each other which have to be enhanced, and another part which is comparatively smooth. The histogram equalization will perform a transformation on a global scale, which may be good for the image in general but not do any good enhancement on particular parts especially if the histogram is dominated by the pixels regarding parts with no detail.

In this project an image called "forestgray.mat" shown in the figure 1, is used.

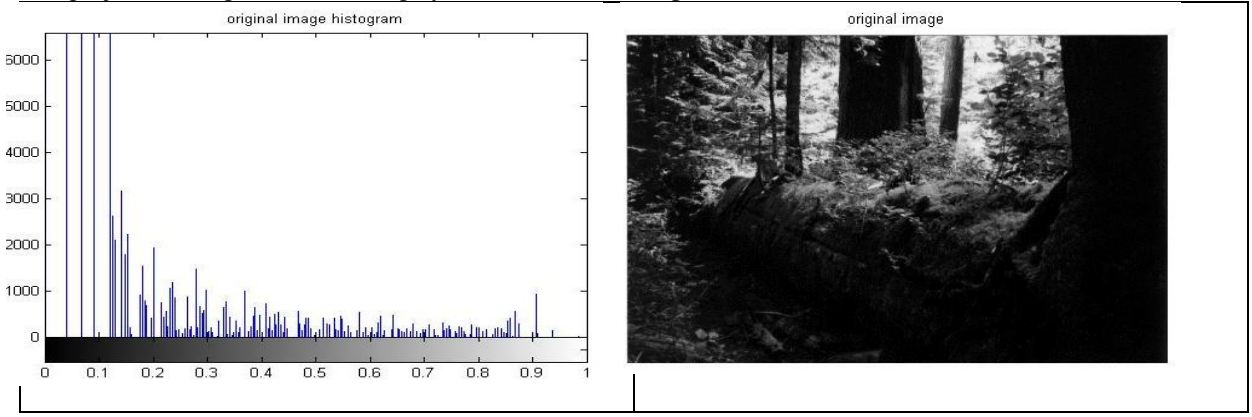



Figure 1) Original image and its histogram.

# 3-Algorithm Description

## 3-1-Local Histogram Equalization

In this method, a window with proper size (L) is defined initially. Afterwards, the window will be applied to the image in a way that the center part of the window will overlap with the pixel in the upper left corner. Assuming that the image is properly zero padded, the histogram of the gray levels of all pixels covered by this window will be generated. Using the achieved histogram and formula (1), the corresponding gray level after applying equalization for the pixel in the center of the image will be computed. In this stage, the window is shifted one pixel to the right and the same procedure will be repeated.

## 3-2- Image enhancement using Histogram Statistical Properties

In this method, initially we create a window of proper size and then we apply it to the image. Assuming that the image is properly zero padded, the histogram of the gray levels of all pixels covered by this window will be generated. Using the achieved histogram and normalizing it by dividing the histogram by the number of pixels in the windowed part, we find the probability of each gray level, , where is the *i:th* gray level. Using

$$m_{S_{xy}} = \sum_{s,t \in S_{xy}} r_{s,t}\, p(r_{s,t}), \quad (2)$$

we find the mean of the gray levels of windowed image pixels. The variance of these gray levels are computed from

$$\sigma_{S_{xy}}^2 = \sum_{s,t \in S_{xy}} \left(r_{s,t} - m_{S_{xy}}\right)^2 p(r_{s,t}). \quad (3)$$

The mean will represent the average gray level of the windowed image which can tell us whether that part of the image is dark or bright. The variance corresponds to the amount of contrast that exists in this image.

Now by comparing the local mean with global mean, we can determine whether the windowed part of the image is dark or bright. To perform the comparison, a coefficient called is defined such that if , where is the global mean of the image, that windowed part of the image is dark.

To evaluate the amount of contrast in this part of the image, the local variance will be compared to the global variance. To do so, a new coefficient is defined, called , such that if , where , is the global variance of the image, the windowed part of the image has low contrast compared to global contrast and as a result, it must be enhanced. If we are interested in enhancing bright parts, the must be greater than one and if we are more interested in enhancing dark parts, must be less than one. However, there are parts on the image which correspond to very low contrast parts and we are not interested in enhancing them. To exclude them from our search, we put a lower bound where is a number smaller than . Finally, if that part satisfies all the criteria, we multiply a constant called to the pixel on the center of the windowed image and then we move the window and we repeat the same procedure. The process described above is summarized as follows if we are interested in enhancing dark parts

$$k(x,y) = \begin{cases} E.r(x,y) & m_{S_{xy}} \le k_0 m_{glob} \; and, k_1 \, \sigma_{glob}^2 \le \sigma_{S_{xy}}^2 \le k_2 \, \sigma_{glob}^2 \\ r(x,y) & otherwise \end{cases}, \quad (4)$$

and if we are interested in enhancing bright parts

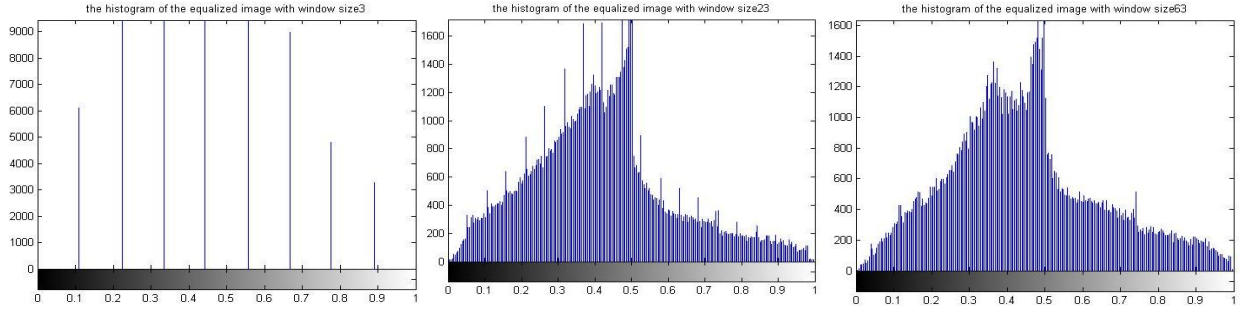
$$k(x,y) = \begin{cases} E.r(x,y) & m_{S_{xy}} \ge k_0 m_{glob} \; and \; k_1 \, \sigma_{glob}^2 \le \sigma_{S_{xy}}^2 \le k_2 \, \sigma_{glob}^2 \\ r(x,y) & otherwise \end{cases}. \quad (5)$$

# 4-Results

## 4-1-Local Histogram Equalization

According to figure 1, this image contains both extremely bright and extremely dark areas. Because of this extreme difference between different parts, even if we apply local histogram, the result will not be very nice. Thus it would seem that histogram equalization will not be appropriate

Figure 2 represents the effect of window size on the local histogram method. Comparing part a) with b), we can see that by increasing the window size from 3 to 23, the histogram has been equalized more and the details on the image are more visible. The reason is that by increasing the window size, we are using more neighborhood pixels which mean we are taking into account the correlation between pixels in a wider range and the transformation will be more effective. Comparing part b) with c), we can see that by increasing the window size from 23 to 63, the histogram has not changed significantly. The reason is that the correlation among pixels very far from each other is negligible. On the other hand, we see two undesirable effects due to increasing the window length. First of all, the simulation time and the number of required computations will increase significantly. Secondly, because we need to perform zero padding around the image with length on each side, by increasing the window length the artifact due to this zero padding will expand to larger part of the image. The pixels with this zero padding artifact become white because there are so many zeros and small gray levels in their window that these pixels will have a large gray level compared to them and after equalization will obtain high values.




a) b) c)

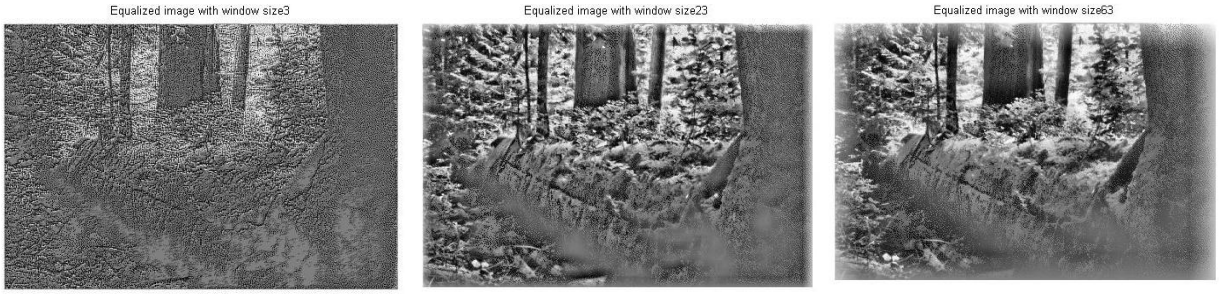



Figure 2) The effect of window size on the result of histogram equalization method a) the output histogram and image for window size=3, b) the output histogram and image for window size=23, c) the output histogram and image for window size=63

## 4-2- Image enhancement using histogram statistical properties

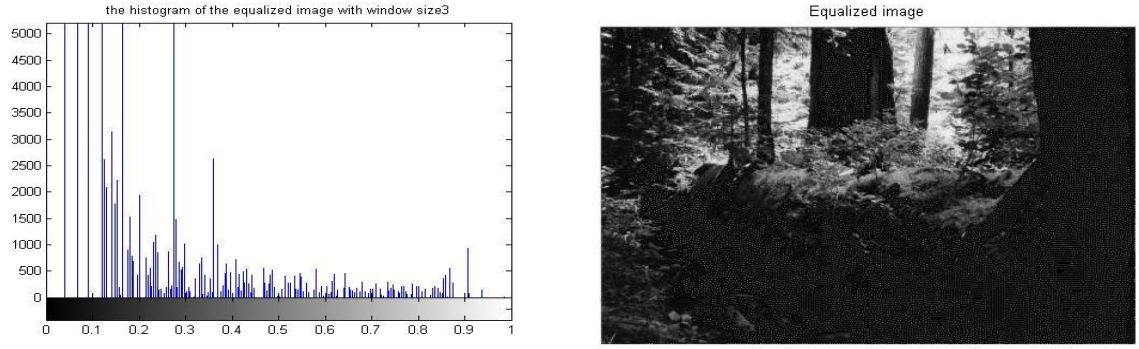



Figure 3) The histogram and the equalized image for E=4, , , window size=3

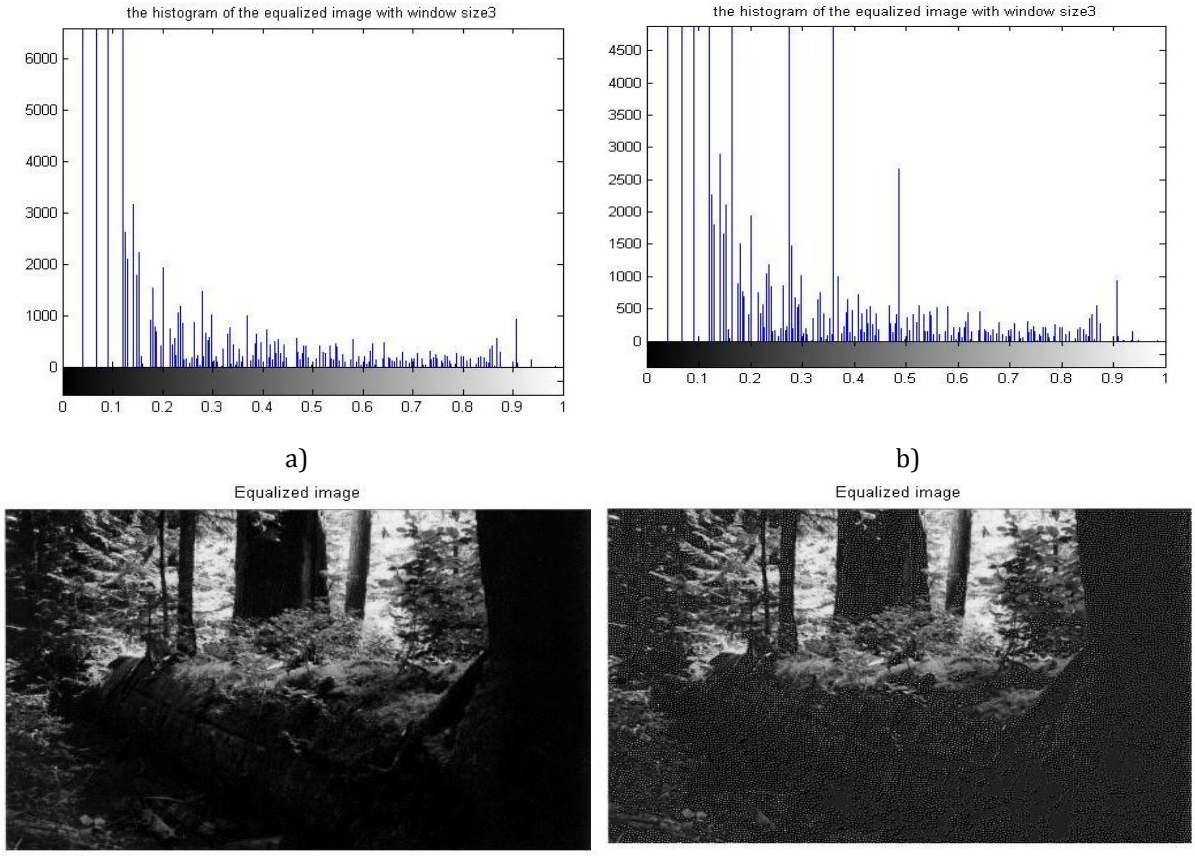



Figure 4) The effect of on the result of second method a) the output histogram and image for , b) the output histogram and image for window

In this part the effect of different parameters on the final histogram is evaluated. Figure 3 represents the enhanced image using the parameters given in the course book. Figure 4 shows the effect of changing on the enhanced image. Comparing it with Figure 3, we can see that by increasing , more parts of the image are enhanced because we detect more parts as dark parts and by decreasing it, we can see the vice versa. Figure 5 shows the effect of changing on the enhanced image. By decreasing it we assume more parts with very low variance must be enhanced and as a result, we might have some dots in a smooth part of the image. By increasing it, very low contrast parts and parts with hidden details will not be enhanced because we assume they are a smooth part of the image and as a result, some details might still remain hidden. Figure 6 shows the effect of changing on the enhanced image. By increasing it, parts with higher contrast will be enhanced and by decreasing it, many parts with low contrast will not be enhanced. Figure 7 shows the effect of changing on the enhanced image. By decreasing it less change in the histogram and image will be seen because we are multiplying the small gray levels with a small number but if we increase it to 10, the previously low contrast and dark parts will be enhanced very much in a way that their gray level will be much higher than the previously bright part of the image and as a result, the image will be darkened. Figure 8 shows the effect of changing window size on the enhanced image. By increasing the window size the artifacts will increase because in order to decide if a pixel must be enhanced or not, we are taking in to account a larger neighborhood which many pixels inside them might have different characteristic to the main pixel and be totally uncorrelated.

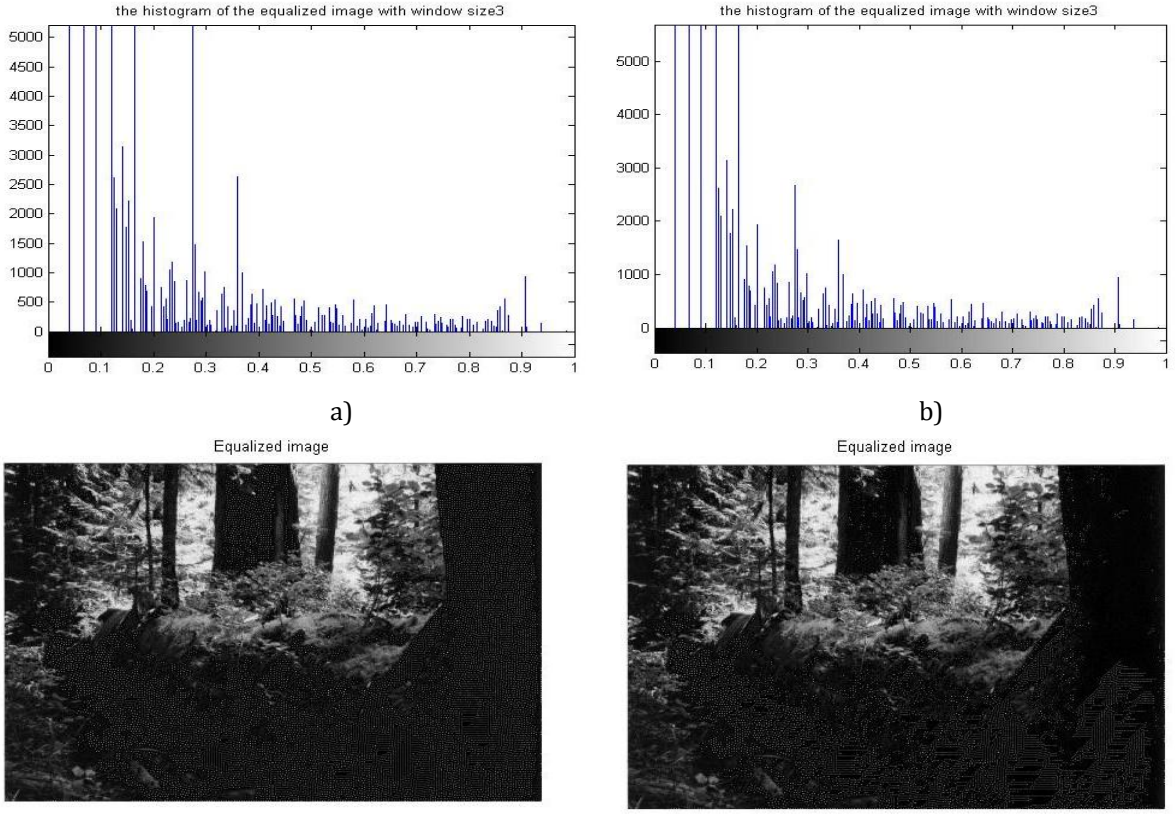



Figure 5) The effect of on the result of histogram equalization method a) the output histogram and image for , b) the output histogram and image for

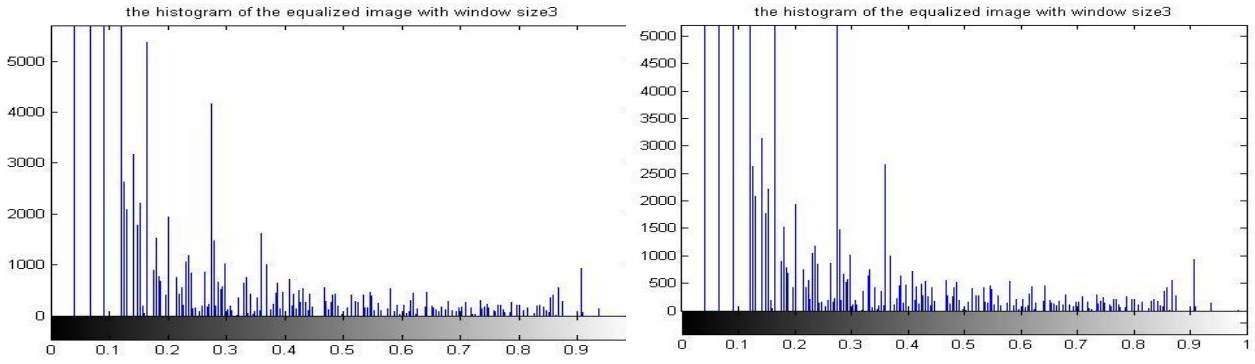
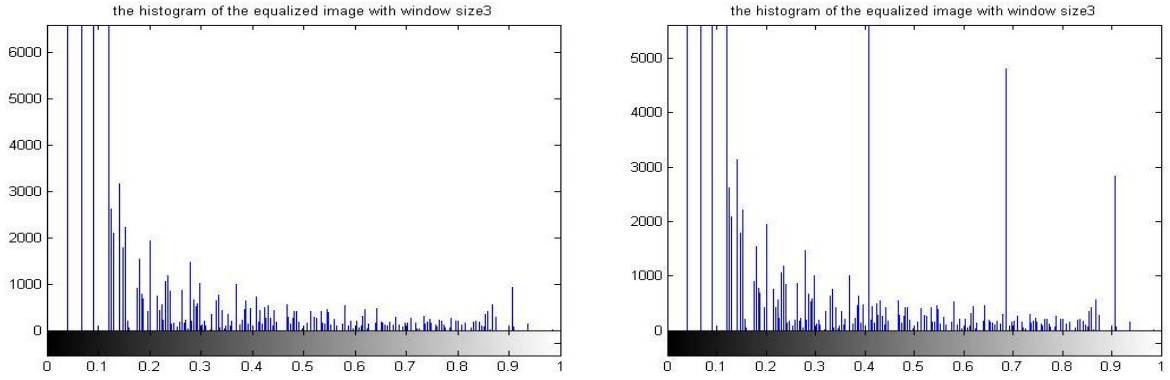



Figure 6) The effect of on the result of histogram equalization method a) the output histogram and image for , b) the output histogram and image for




a) b)

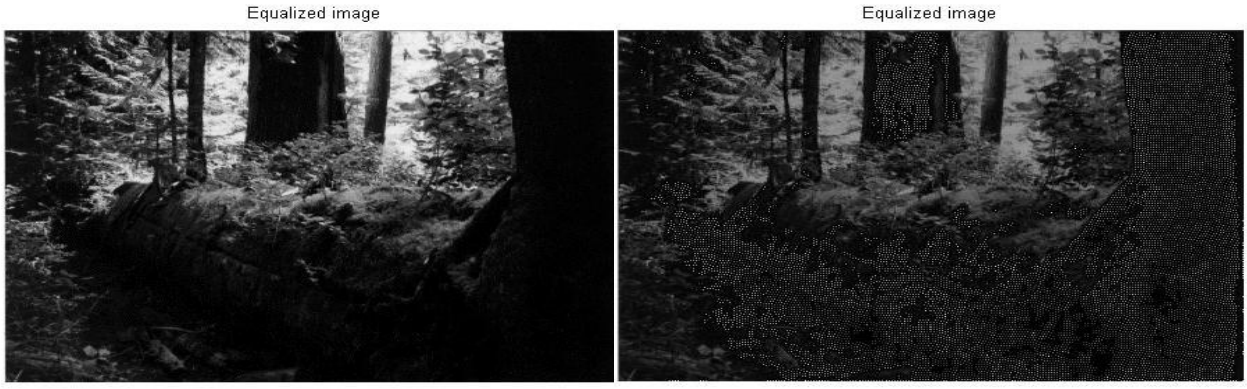
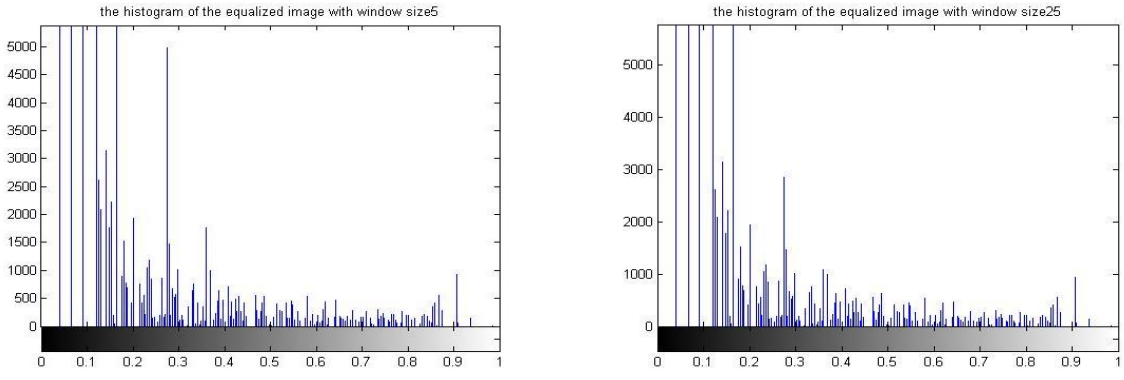



Figure 7) The effect of E on the result of histogram equalization method a) the output histogram and image for E=1, b) the output histogram and image for E=10




a) b)

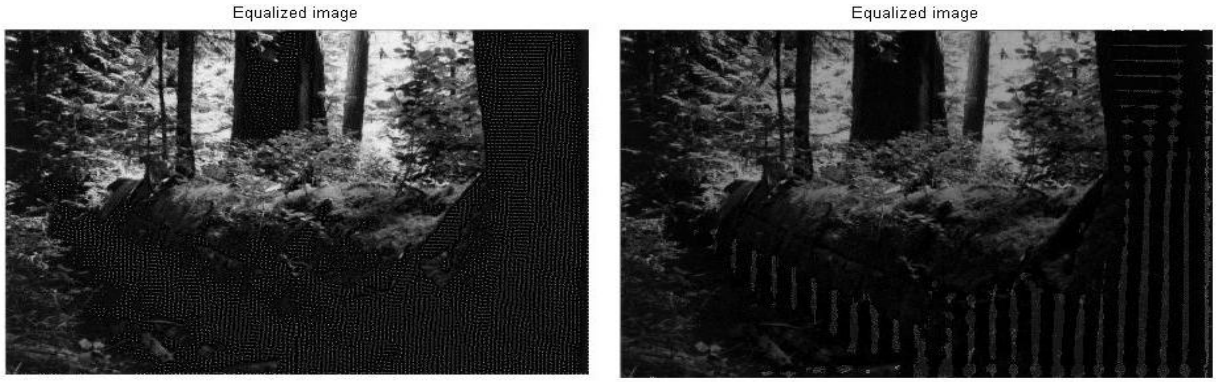



Figure 8) The effect of window size on the result of histogram equalization method a) the output histogram and image for window size=5, b) the output histogram and image for window size=25

## 4-3-Extensions

### 4-3-1) Improving the Image

As it was mentioned earlier, the interval between the largest and smallest pixel in this image is very large and as a result, histogram equalization might not result in good images. However, if we take a logarithm from this image, the interval will become smaller and as a result, it will not have very high extremes and by doing histogram equalization, we will have good images

## b) A Proposal to Decrease the Simulation Time

One of the disadvantages of this method is that we need to compute the local histogram times where $N$ and $M$ are the number of rows and columns in the image respectively, which will be time consuming especially when the window length increases. To decrease the simulation time the following method is suggested. In this method, we create a vector with size equal to the number of gray levels in the picture (e.g., 256 elements for an 8-bit image), called S and in the first time, we put the number of pixels having each of these gray levels on the corresponding location in the vector (e.g., if we have 3 pixels with gray level 134, we save 3 on the $134^{th}$ element of this vector). We save the original S in another vector and then when the window is shifted to the next part, we make proper alterations to this vector given by

We repeat this procedure until we reach the end of the row. Now if we are doing roaster scanning, we substitute with the original S and we perform (5) by moving the window one row downward. In this stage, we replace original S with . We do so in order to use it when we go to the next row. This method will reduce the number of required computations significantly.

## 4-4-Discussion

**Enhancement using local histogram equalization:** By doing histogram equalization on small areas of an image, one can deal with many problems. Noisy parts will not affect the whole image. Bright and dark areas are often equalized separately, so that they will not be affected by the other parts. Areas with a high amount of details are dealt with separately. A disadvantage with this method is that it creates “snow-like” features around edges. It also reveals high contrast noise. Local histogram equalization can be used when the aim is to reveal some hidden details in the image specially when these hidden parts have gray levels very similar to neighboring parts or are in a way that they are not really affecting the global histogram of the image.

**Enhancement using histogram statistical properties:** This method uses statistical properties to determine whether the area is dark or low contrast and needs to be enhanced. The advantage with this method is that you only manipulate parts of the image that you want to enhance and so eventual artifacts will only be created in these parts. In the dark areas of the tree, some strong artifacts appear where the bark of the tree should be visible. As the window size is increasing, large striped artifacts appear. The same effect appears in Figure 3.26 in the course book [1]. In the edge between the white filament and the black background artifacts are reveled. Enhancement using histogram statistical properties is good when you want to enhance areas which are too dark or bright to see any details but you want to let the good parts unchanged. For instance, if an object is properly viewed in the foreground of the image whereas there are hidden parts in the dark back ground, or other examples where the statistical properties of the image varies locally and we only want to enhance parts of the image, we use this method.

# 5-Appendix

**Program 1: Local histogram enhancement**

```
clear all
close all
clc

% THIS PROGRAM PERFORMS LOCAL HISTOGRAM EQUALIZATION ON THE IMAGE.

%%%%%%% Parameters %%%%%%%%%%
Num_Level=256;    % the number of gray levels of the image.
Window_Size=3;    % local histogram window size
plotflag=1;       % If "plotflag" is zero, it is not plotting.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% LOADING AND ZEROPADDING IMAGE %%%%%%%%
load('forest.mat')  % Loading the image.
Original_Im=forestgray; % Putting the image into a matrix.
% The image is padded with zeros in a frame around it, which is half the
% size of the windows used, rounded down to nearest integer.
Zero_Padded_Im=zeros(Window_Size-1+size(Original_Im,1),Window_Size-
1+size(Original_Im,2));
Zero_Padded_Im(floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,1),floo
r(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,2))=Original_Im;

%%%% MOVING THE WINDOW THROUGH THE IMAGE %%%%%%
% Moving through the rows of the image
for Row=floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,1)
    for Col=floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,2)
        % Creating a window with the pixel (Row,Col) in the centre and the
        % size, "Window_Size".
        Windowed_Image=Zero_Padded_Im(Row- floor(Window_Size/2):...
Row+floor(Window_Size/2),Col-floor(Window_Size/2):Col+floor(Window_Size/2));
        % The pixel in the centre of the window is assiged to the variable
        % "Input_Graylevel".
        Input_Graylevel=Zero_Padded_Im(Row,Col);%%%the gray level of the pixel in
the middle of the window that we like to do the histogram equalization on
        % The function "Hist_Eq" is used to perform a histogram
        % equalization on the window. The outputs are the equalized centre
        % pixel "Out_Graylevel" and the transformation function "Transform".
        [Out_Graylevel, Transform] =
Hist_Eq(Windowed_Image,Input_Graylevel,Num_Level );
        % The pixel (Row,Col) is replaced with "Out_Graylevel".
        Zero_Padded_Im(Row,Col)=Out_Graylevel;
    end
end

%%%%%%% CUTTING THE ZERO PADDED PART %%%%%%%%%%%%%
Equalized_Im=Zero_Padded_Im(floor(Window_Size/2)+1:floor(Window_Size/2)+size(Origin
al_Im,1),floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,2));


%%%%%%%%%%%% PLOTTING %%%%%%%%%%%%%%%%
if (plotflag)
    figure
    imshow(Original_Im, [])
    title('original image')

    figure
    imhist(Equalized_Im,Num_Level)
    Sentence=strcat('the histogram of the equalized image with window size
',num2str(Window_Size));
    title(Sentence)

    figure
    imshow(Equalized_Im, [])
    title('Equalized image')
end
```

**Program 2: Function for histogram enhancement**

```
function [Out_Graylevel, Transform] = Hist_Eq(
Windowed_Image,Input_Graylevel,Num_Level )

% This function is used to perform a histogram equalization on a specific part
% of an image, "Windowd_Image". "Input_Graylevel" is the gray level of the
% centre of "Windowed_Image". "Num_Level" is the number of gray levels of
% the image. "Out_Graylevel" is the gray level of the centre of
% "Windowed_Image" after histogram equalization is performed. "Transform"
% is the tranformation function.

% Creating the histogram of the image.
[Im_Hist IM_Index]=imhist(Windowed_Image,Num_Level);
% Creating cumulative density function, which is the transformation
% function.
Transform=cumsum(Im_Hist)./prod(size(Windowed_Image));
% Using the transformation function to find the output level.
% "Index" is the corresponding index of the transfer function.
Index=round(Input_Graylevel*(Num_Level-1))+1;
Out_Graylevel=Transform(Index);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

**Program 3: Program for image enhancement using histogram statistics**

```
clear all
close all
clc
% THIS PROGRAM PERFORMS IMAGE ENHANCEMENT USING HISTOGRAM STATISTICS.
%%%%%%% Parameters %%%%%%%%%%
Num_Level=256;    % the number of gray levels of the image.
Window_Size=61;   % Size of the windows which local histogram equalization is
performed.
plotflag=1;       % If "plotflag" is zero, it is not plotting.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% LOADING AND ZEROPADDING IMAGE %%%%%%%%
load('forest.mat')  % Loading the image.
Original_Im=forestgray; % Putting the image into a matrix.
% The image is padded with zeros in a frame around it, which is half the
% size of the windows used, rounded down to nearest integer.
Zero_Padded_Im=zeros(Window_Size-1+size(Original_Im,1),Window_Size-
1+size(Original_Im,2));
Zero_Padded_Im(floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,1),floo
r(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,2))=Original_Im;
%%%% DEFINING THE PARAMETERS %%%%%%
E=4;            % The factor with which the parts of the image, fullfilling the
criterias of the mean and standard deviation, is enhaced.
k0=0.4;          % First criteria: The mean of the window must be smaller than k0
multiplied with the global mean
k1=0.02;         % Second criteria: The standard deviation of the window must be
between k1 and k2 multiplied with the local mean.
k2=0.4;

% The function "Mean_Eq" is used to determine the global mean "Mean_Glob" and
standard
% deviation "StdDev_Glob" of the zeropadded image.
[Mean_Glob, StdDev_Glob] = Mean_Eq( Zero_Padded_Im,Num_Level );
 %%%% moving the window through the image%%%%%%

% Moving through the rows of the image
for Row=floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,1)
    Row  % The Row number is displayed to monitor the progress of the process.
    % Moving through the columns of the image.
    for Col=floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,2)
        % Creating a window with the pixel (Row,Col) in the centre and the
        % size, "Window_Size".
        Windowed_Image=Zero_Padded_Im(Row-
floor(Window_Size/2):Row+floor(Window_Size/2),Col-
floor(Window_Size/2):Col+floor(Window_Size/2));
        % The pixel in the centre of the window is assiged to the variable
        % "Input_Graylevel".
```

```
        Input_Graylevel=Zero_Padded_Im(Row,Col);
        % Determining the local mean and standard deviation
        [local_Mean, local_StdDev] = Mean_Eq( Windowed_Image,Num_Level );
        % Multiplying the centre pixel of the window with the parameter "E"
        % if the criterias are fulfilled. Otherwise, leave the pixel
        % unchanged.
        if (local_Mean<=k0*Mean_Glob & local_StdDev<=k2*StdDev_Glob &
local_StdDev>=k1*StdDev_Glob)
         Zero_Padded_Im(Row,Col)=E*Input_Graylevel;
        else
        Zero_Padded_Im(Row,Col)=Input_Graylevel;
        end
    end
end
%%%%%%% CUTTING THE ZERO PADDED PART %%%%%%%%%%%%%
Equalized_Im=Zero_Padded_Im(floor(Window_Size/2)+1:floor(Window_Size/2)+size(Origin
al_Im,1),floor(Window_Size/2)+1:floor(Window_Size/2)+size(Original_Im,2));


%%%%%%%%%%%% PLOTTING %%%%%%%%%%%%%%%%
if (plotflag)
    figure
    imshow(Original_Im, [])
    title('original image')

    figure
    imhist(Equalized_Im,Num_Level)
    Sentence=strcat('the histogram of the equalized image with window size
',num2str(Window_Size));
    title(Sentence)

    figure
    imhist(Original_Im,Num_Level)

    title('original image histogram')

    figure
    imshow(Equalized_Im, [])
    title('Equalized image')
end
```

**Program 4: Function for calculating histogram mean and standard deviation**

```
function [Mean, StdDev] = Mean_Eq( Windowed_Image,Num_Level )

% This function is finding the mean gray level and the standard deviation of
% the gray levels of an image.

% Creating the histigram "Im_Hist" and setting up a vector of the gray
% levels "Im_index" of the image.
[Im_Hist Im_Index]=imhist(Windowed_Image,Num_Level);
% Creating the probability density function.
Im_Prob=Im_Hist./prod(size(Windowed_Image));
% Calculating the mean and standard deviation of the image.
Mean=sum(Im_Index.*Im_Prob);
StdDev=sqrt(sum(((Im_Index-Mean).^2).*Im_Prob));
```