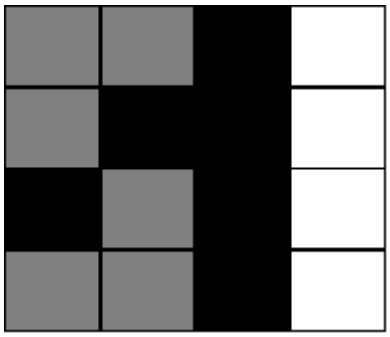
Exercise Chapter 2 – Computing a histogram

In this exercise you will learn to use the image histogram function ***imhist*** which creates and plots the image histogram. A simple example is introduced in order to understand what is a histogram. Then you will compare different image histograms (grayscale and color images).

1 – Use the Matlab help (command ***help*** or ***helpwin***) to display the description of the Matlab function ***imhist***.

2 – Create the matrix *im1* such as each element *im1(i, j)*, (i, j) ∈ [1, 4]², specifies the gray level of the pixel (i, j) in the following image:



Use the following values:

- Black pixel intensity = 0 ;
- Gray pixel intensity = 127/255 ;
- White pixel intensity = 1.

Display *im1* with the ***imagesc*** function and check your coefficients visually (you can build an adequate LUT to display your image in gray levels. To apply this LUT, use the ***colormap*** function).
Plot the ***im1*** image histogram with the command: ***imhist(Im1,3)***. Analyze the result.

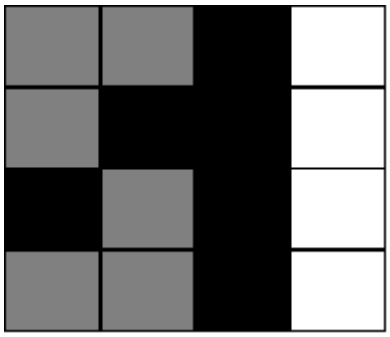
3 – Load into your working folder the two grayscale images « *FRUIT_LUMI* » and « *ISABE_LUMI* ». Open these two images with the ***imread*** function. Display and compare the histograms of the two images.

4 – Load and open the color image « *MANDRILL* ». Use the ***imhist*** function and visualize plane by plane the histograms of the RGB planes for this color image.

Exercise solution: Computing a histogram

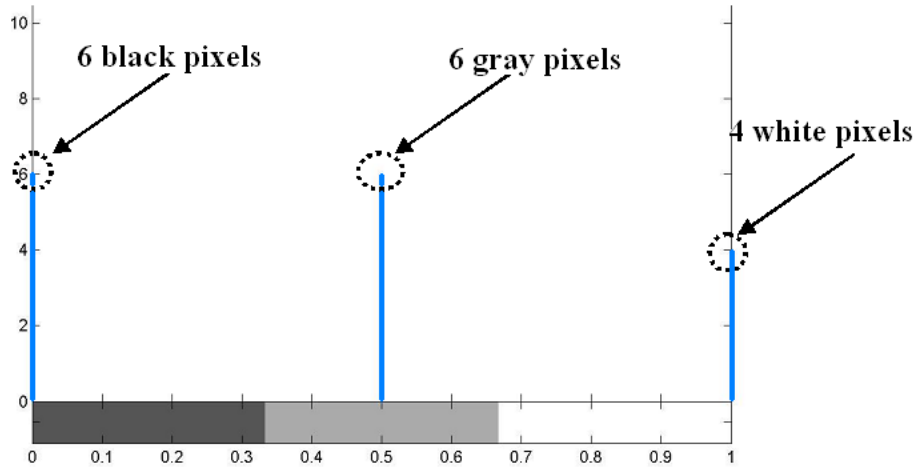
1 – The ***imhist*** function computes and plots grayscale image histograms. The ***imhist*** function can only be used plane by plane for color images.

2 – Here are the commands to plot the histogram of the 4×4 image:



```
im1=[127/255 127/255 0 1;127/255 0 0 1;0 127/255 0 1; 127/255 127/255 0 1]
% Build the LUT to display in gray levels
r=[0 0.5 1];
g=[0 0.5 1];
b=[0 0.5 1];
map=[r' g' b'];
imagesc(im1);
colormap(map) ;
% Plot the image histogram
figure
imhist(im1,3) ;
```

Here is the plotted histogram:

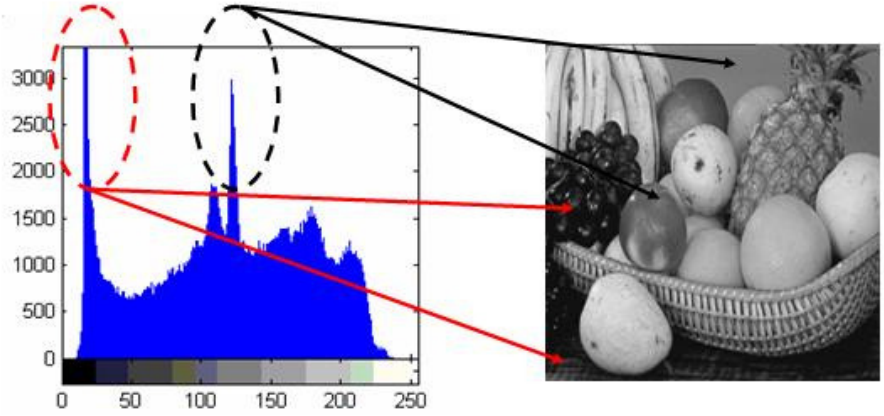


The histogram displays the number of pixels for each gray level. There are thus 6 black pixels, 6 gray pixels, and 4 white pixels. You can easily visualize this result on the original 4×4 image.

3 – Let us read the grayscale images *FRUIT_LUMI.BMP* and *ISABE_LUMI.BMP* with the command ***imread***. We can thus build the histograms for these grayscale images:

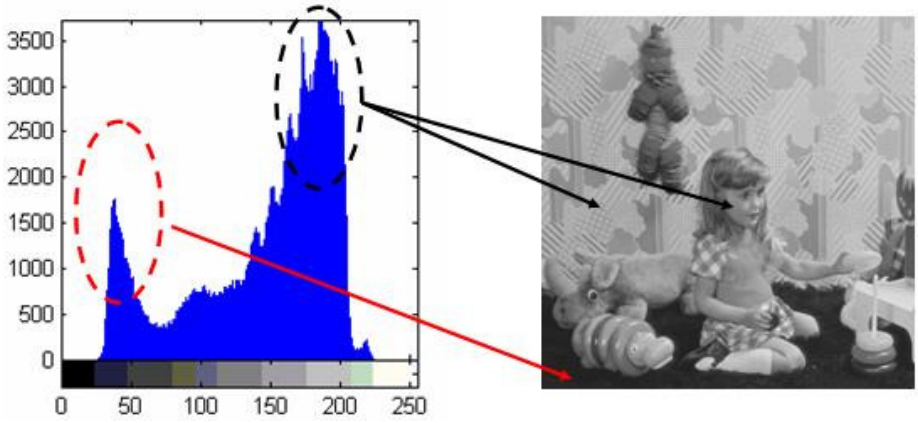
```
I = imread('FRUIT_LUMI.BMP') ;
J = imread('ISABE_LUMI.BMP') ;
imhist(I);
imhist(J);
```

The two peaks of pixel populations in the *FRUIT_LUMI* image histogram correspond to the darkest image zones (ground, grape, etc.) and to the zones which have a middle gray level (intensity close to 127 for apples, background, etc.).



Histogram and FRUIT_LUMI image

Here is the histogram of *ISABE_LUMI* image:

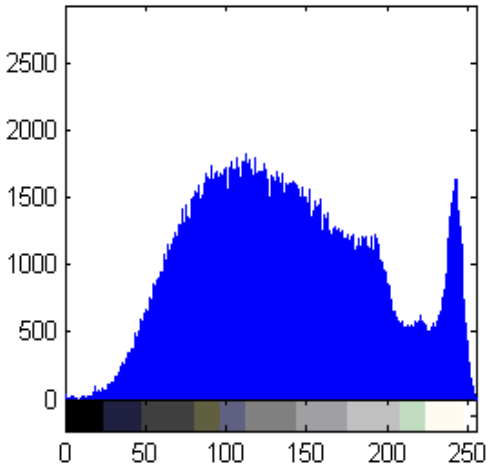


Histogram and ISABE_LUMI image

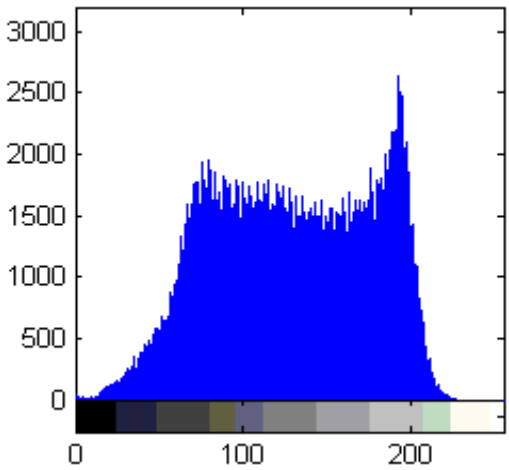
Here the most important peaks of population correspond to the floor shown in the image (the darkest zone) and to the background of the image (the lightest zone).

4 – In the case of color images such as *MANDRILL.BMP* image, we read the image file with the ***imread*** function then we compute the histogram plane by plane (there are three achromatic planes: Red, Green, and Blue).

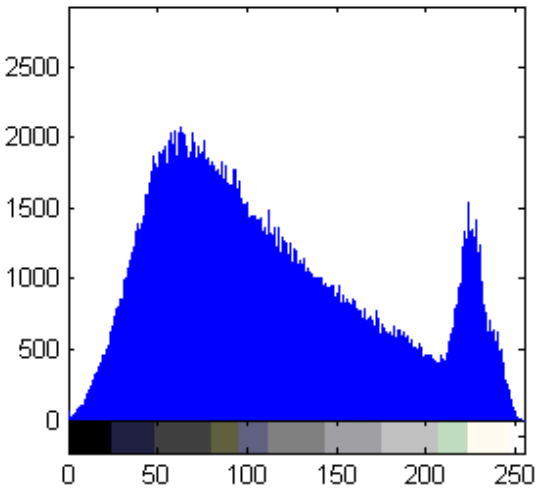
```
I = imread('MANDRILL.BMP') ;
imhist(I(:,:,1));  % first plane: Red
imhist(I(:,:,2));  % second plane: Green
imhist(I(:,:,3));  % third plane: Blue
```



MANDRILL.BMP (first plane: Red)



MANDRILL.BMP (second plane: Green)



MANDRILL.BMP (third plane: Blue)