Computer Image Processing

Lecture 4

Histogram

Arithmetic operations

Histogram

A **histogram** is one of the graphical ways of presenting the distribution of some feature.

If we consider pixel values, histogram presents an information about number of pixels that have a given value in the picture.

The calculation of histogram components is performed in the following way:

$$n_i = \sum_{x=1}^{M} \sum_{y=1}^{N} g_i(x, y)$$

where:

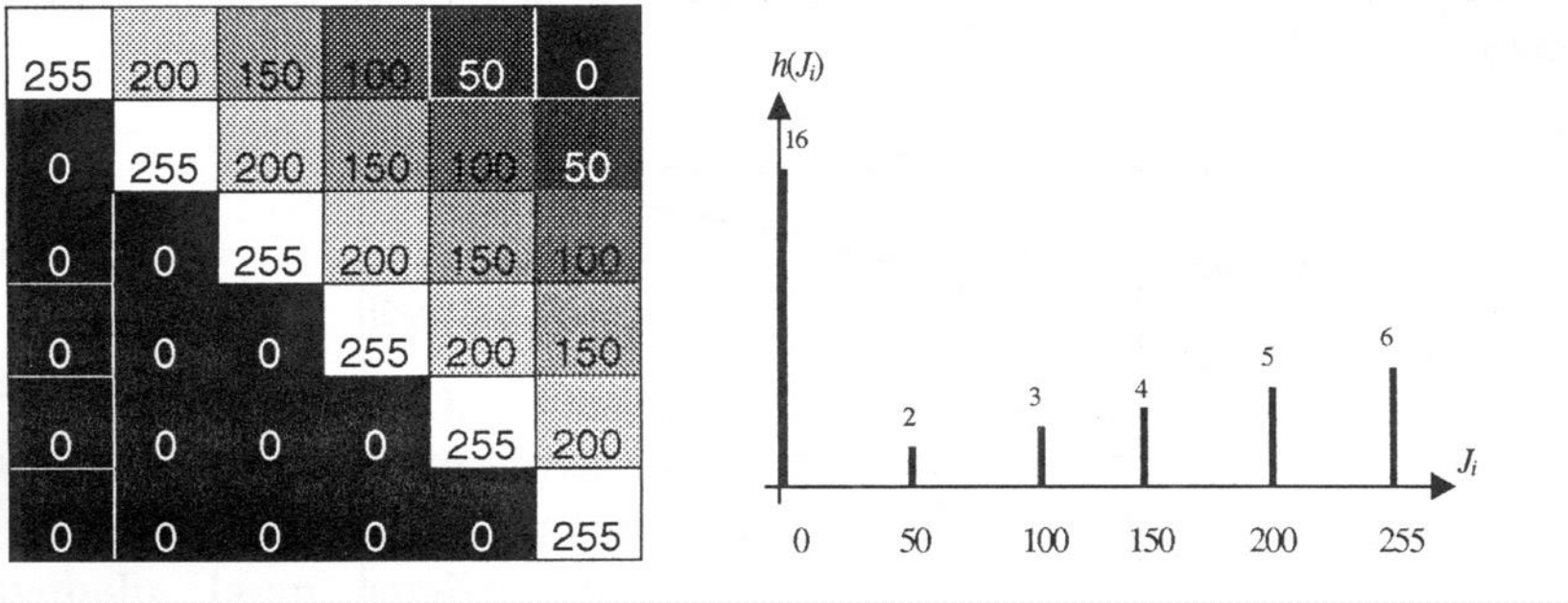
n_i – number of pixels that have a given value i (J_i),

M – horizontal size of the image,

N – vertical size of the image,

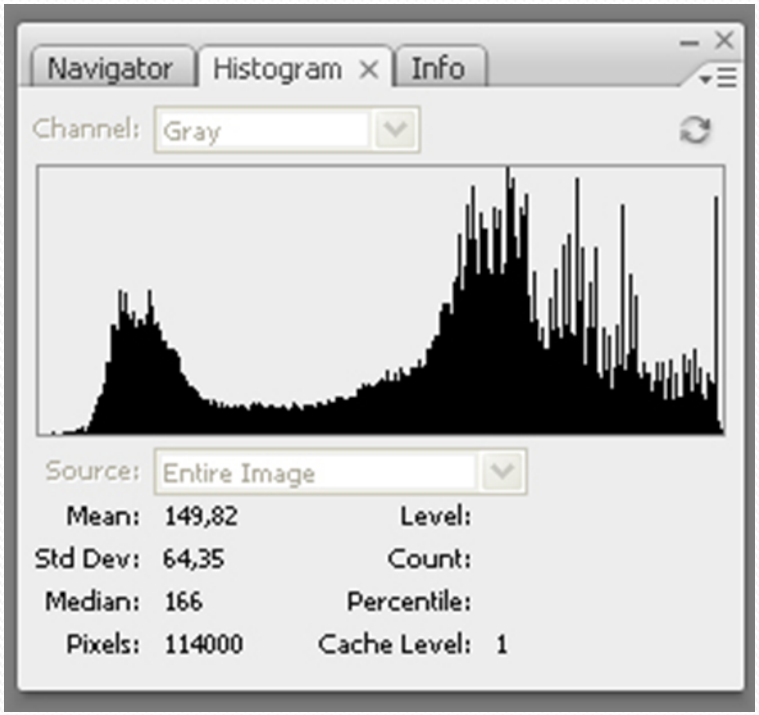
$$g_i(x, y) = \begin{cases} 1 & \text{if } J(x, y) = i \\ 0 & \text{else} \end{cases}$$

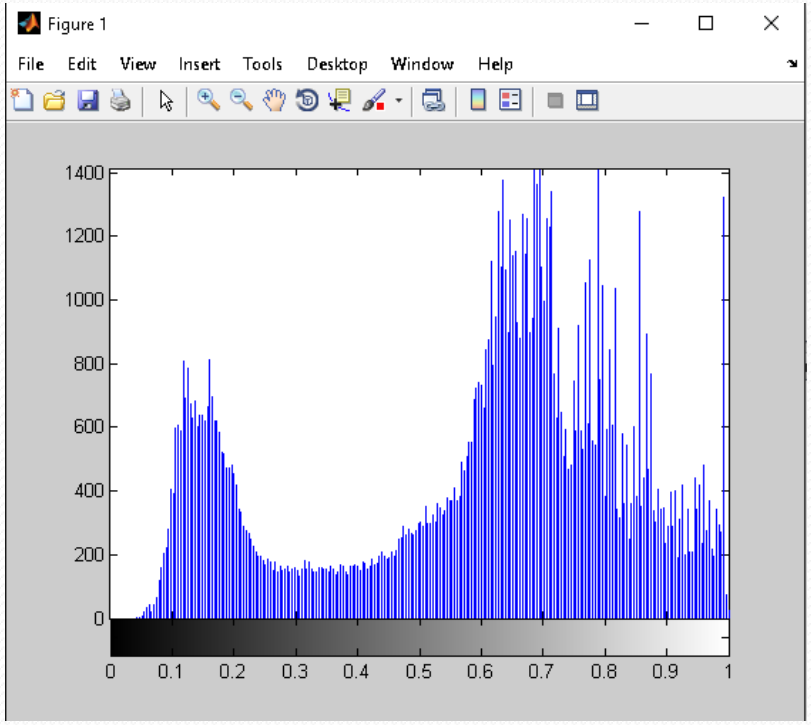
Histogram



Sample image and its histogram.

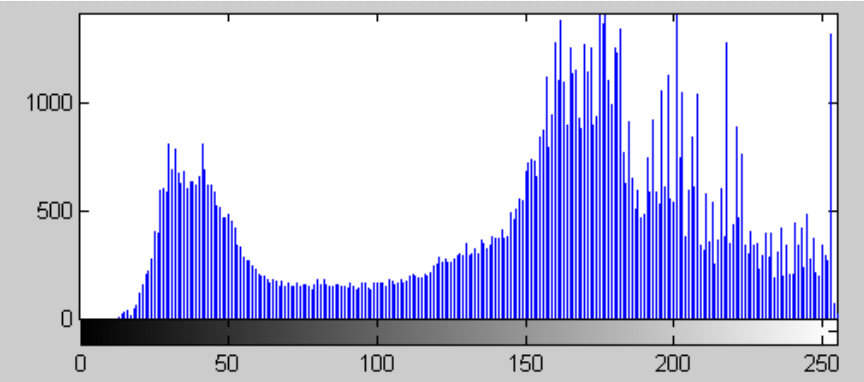
Histogram



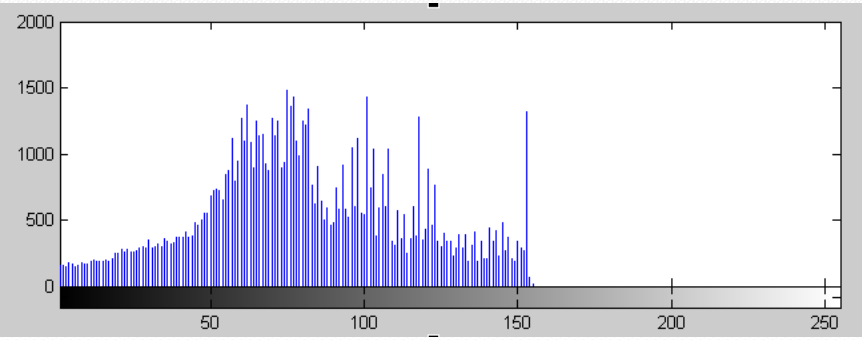


Histogram in Adobe Photoshop and Matlab programs.

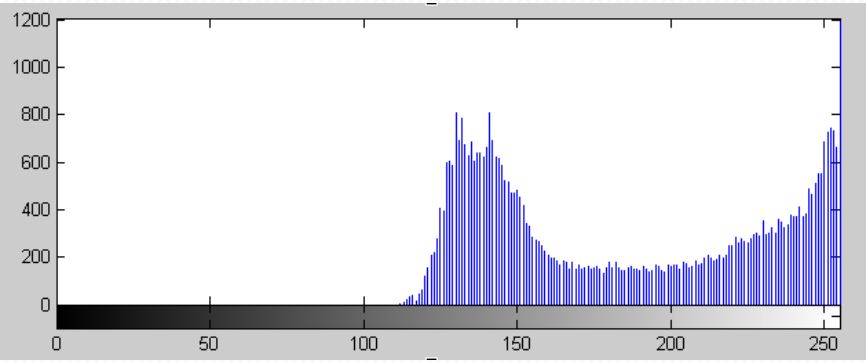
Histogram



Histogram of the correct image - saturating the full range of brightness levels

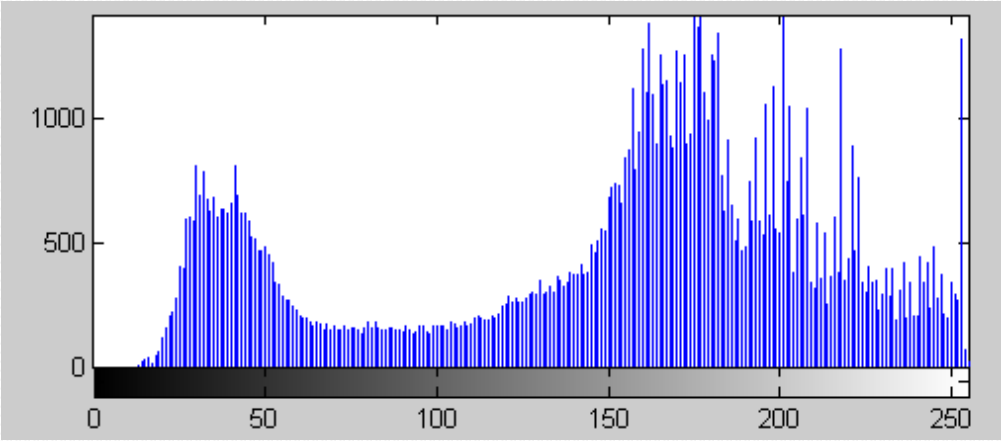


Histogram of an image that is too dark - mainly low-bright pixels in the image, no high-brightness pixels

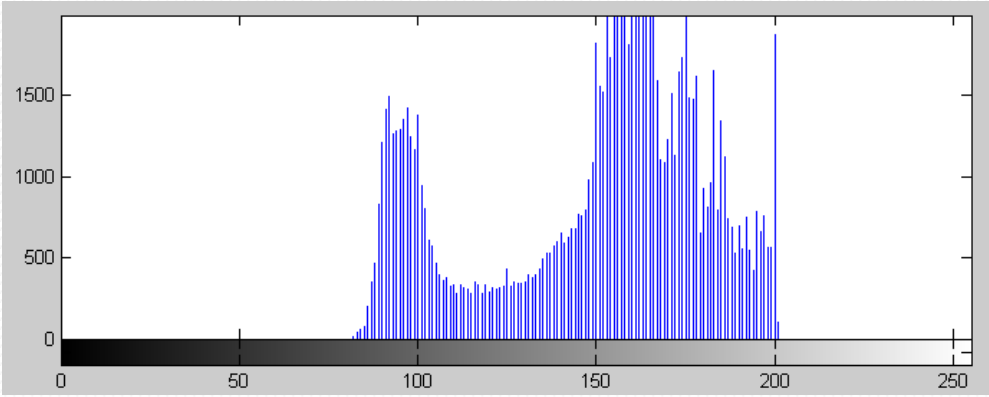


Histogram of an image that is too light - mainly high-bright pixels in the image, no low-bright pixels

Histogram



High contrast image histogram



Low contrast image histogram

Operations on the histogram

Extending the brightness range (stretching the histogram)

Transformation is performed when the pixel value range of the image does not cover the entire available range. The result of this operation is increasing the contrast of the image, because its pixels with minimum and maximum values will achieve available extreme values (0 and 255), while between them the distances will increase.

$$J_{out}(x,y) = \frac{255}{J_{\max} - J_{\min}} \cdot (J(x,y) - J_{\min})$$

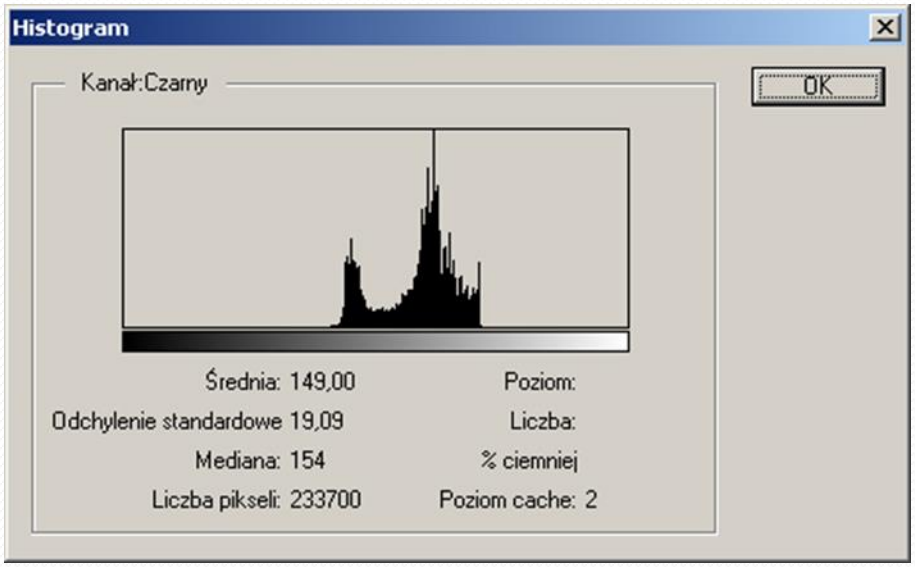
$$\text{for} \quad J_{\min} \le J(x,y) \le J_{\max}$$

Operations on the histogram

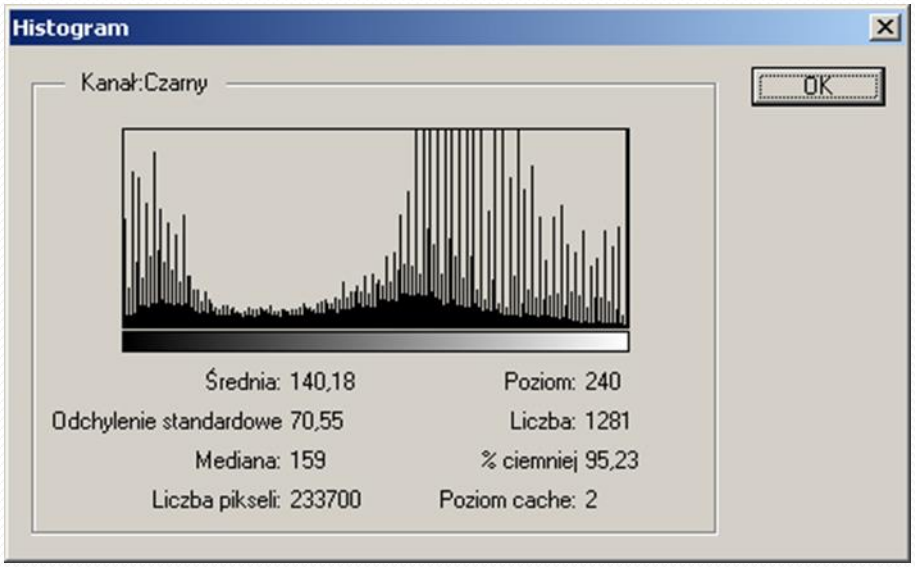
Source low-contrast image

Image after histogram stretching

Operations on the histogram

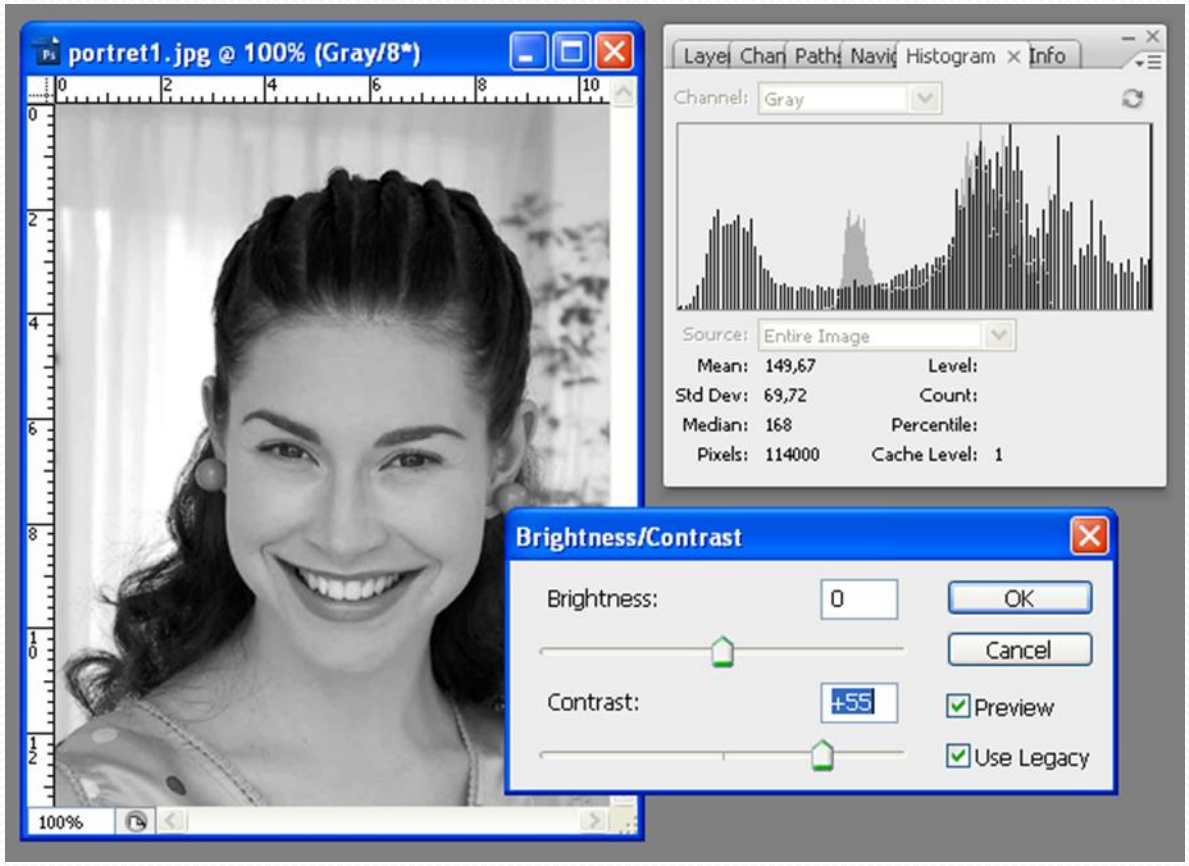


Histogram of a low-contrast image



Histogram after stretching

Operations on the histogram



Stretching the histogram

Increasing contrast

Using the entire range of brightness levels

Operations on the histogram

Source low-contrast image

Image after histogram stretching

Operations on the histogram

Histogram equalization

The operation is based on equalizing the gray levels in a way that the histogram would be as flat as possible, i.e. all gray levels have a similar number of points. This operation allows emphasizing these details in the picture, which due to the small contrast are hardly visible, because the human eye is better able to recognize objects when they are separated from each other in terms of brightness.

Pixels with luminosities that often appear in the image are more clearly distinguishable because they are usually qualified to different levels in the resulting image.

Pixels with luminosities that occur rarely can be classified to one level, which results in the loss of some data.

Operations on the histogram

Histogram equalization

Example:

Consider the image, size 64 x 64 pixels

$s = 4096$ pixels

$k = 8$ levels of brightness

h_n - probability of occurence pixel with the value n

n	h_n	h_n/s	$D(n)$
0	790	0,19	0,19
1	1023	0,25	0,44
2	850	0,21	0,65
3	656	0,16	0,81
4	329	0,08	0,89
5	245	0,06	0,95
6	122	0,03	0,98
7	81	0,02	1,00

Operations on the histogram

Histogram equalization

To create a resulting image the *Look Up Table* could be prepared
For $i = 1$ intensity level, the LUT value is calculated as:

$$LUT(1) = \frac{0{,}44 - 0{,}19}{1 - 0{,}19} \cdot (8 - 1) \approx 2$$

Full Look Up Table

x_i	0	1	2	3	4	5	6	7
y_i	0	2	4	5	6	7	7	7

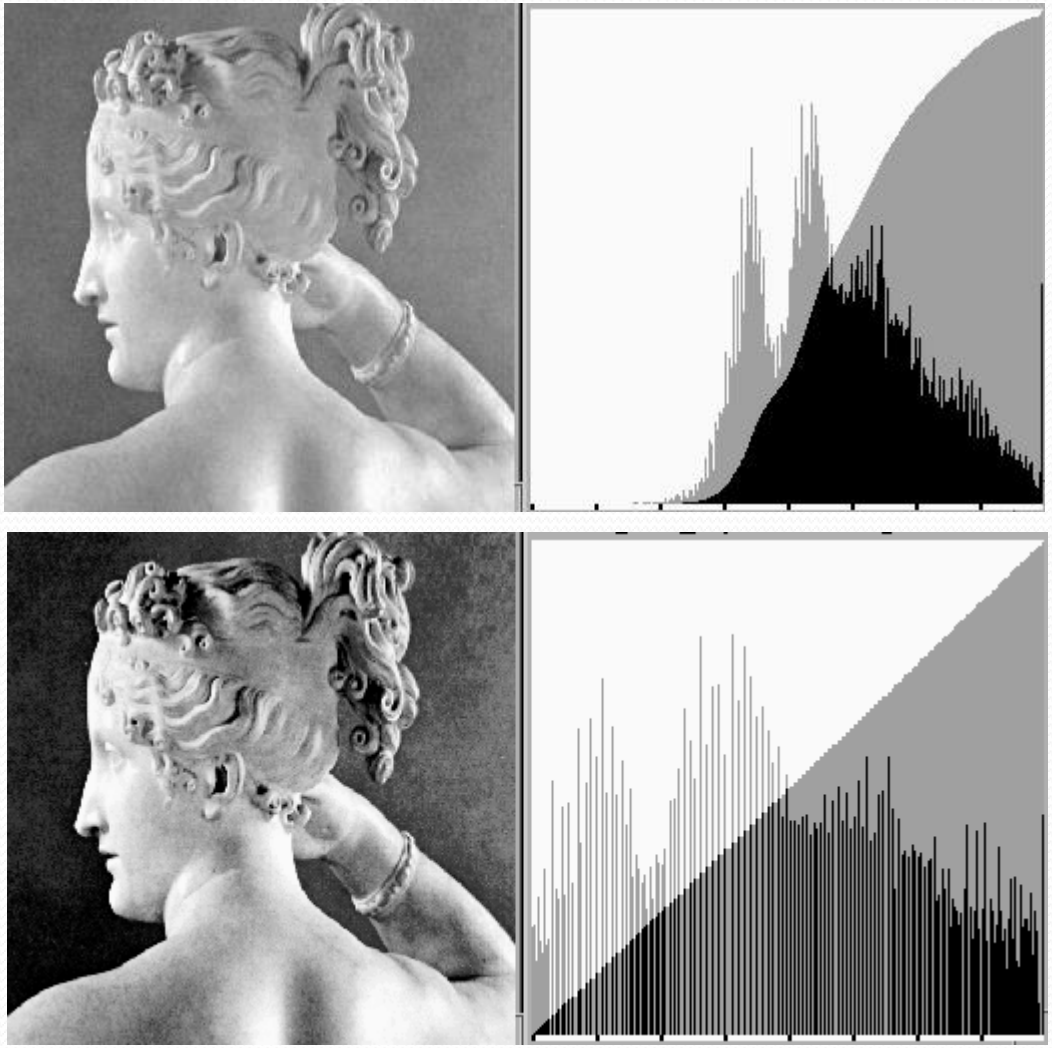
x_i - intensity levels in original image
y_i - intensity levels in resulting image

Operations on the histogram

J_i	n_i	$h(J_i)$
$J_0 = 0$	790	0,19
$J_1 = 1/7$	1023	0,25
$J_2 = 2/7$	850	0,21
$J_3 = 3/7$	656	0,16
$J_4 = 4/7$	329	0,08
$J_5 = 5/7$	245	0,06
$J_6 = 6/7$	122	0,03
$J_7 = 1$	81	0,02

s_k	$\approx s_k$	n'_i
$s_0 = T(J_0) = 0,19$	1/7	790
$s_1 = T(J_1) = 0,19+0,25 = 0,44$	3/7	1023
$s_2 = T(J_2) = 0,65$	5/7	850
$s_3 = T(J_3) = 0,81$	6/7	985
$s_4 = T(J_4) = 0,89$	6/7	
$s_5 = T(J_5) = 0,95$	1	448
$s_6 = T(J_6) = 0,98$	1	
$s_7 = T(J_7) = 1$	1	

Operations on the histogram

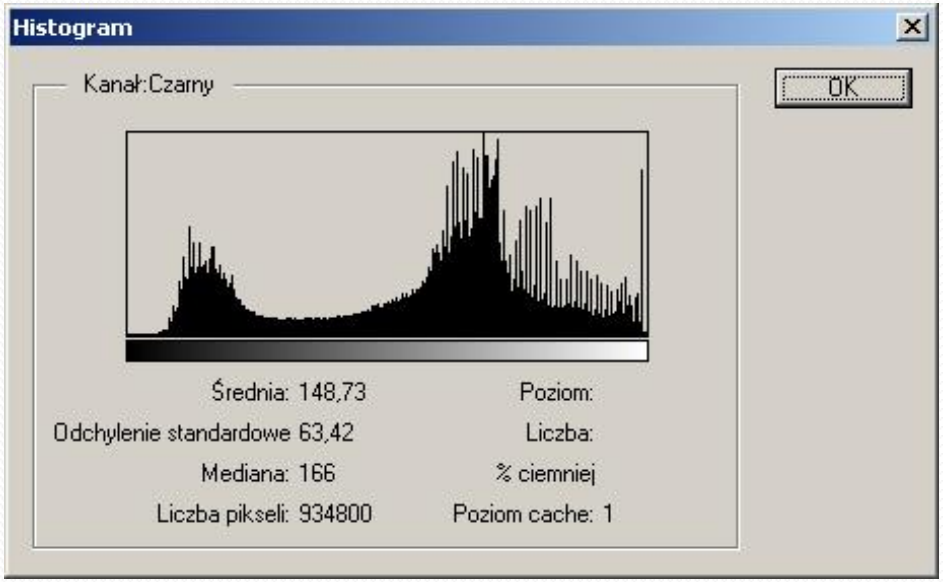


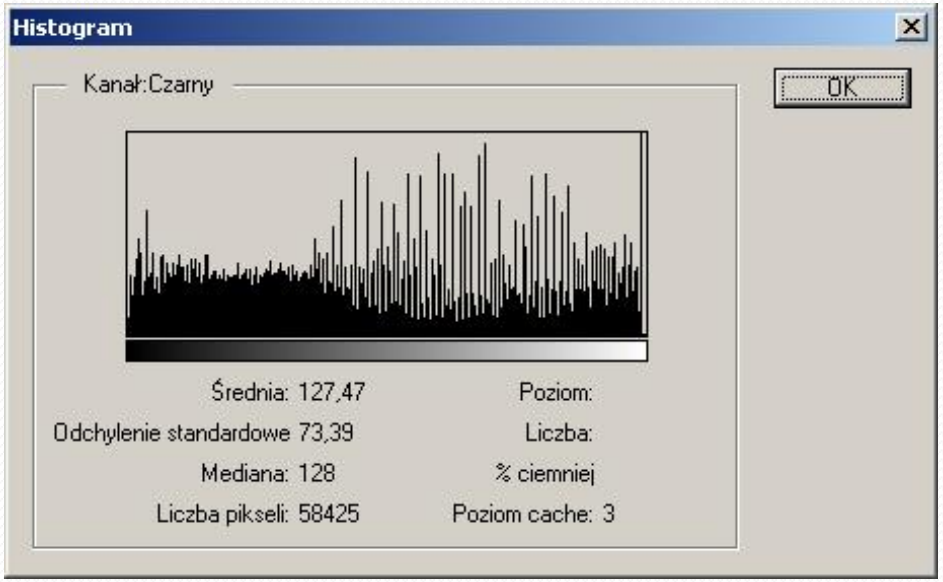
Operations on the histogram

Source image

Image after histogram equalizing

Operations on the histogram





Histogram of an original image

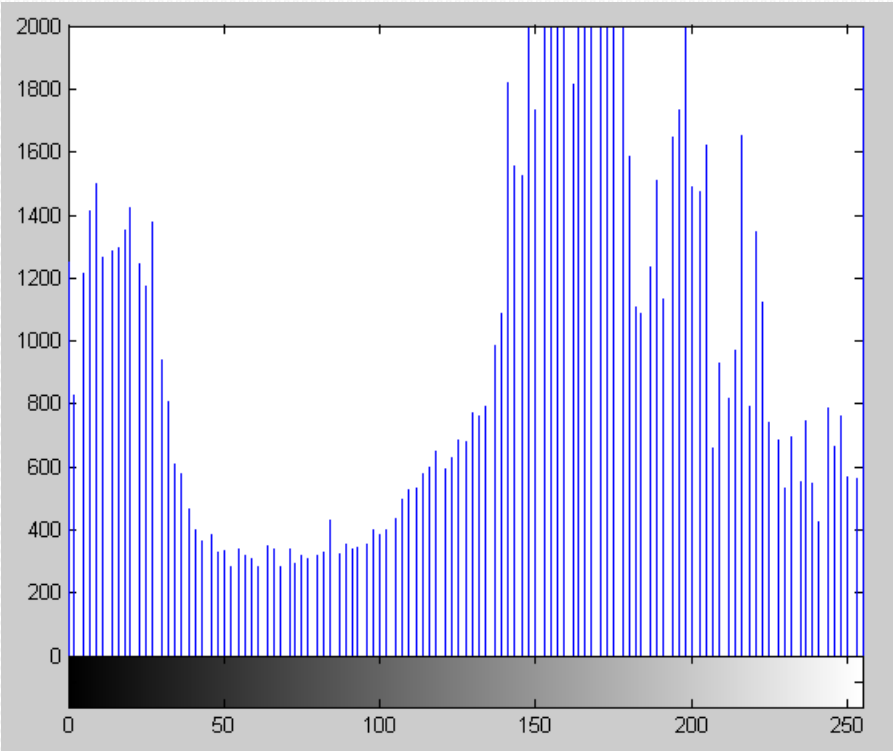
Histogram after equalizing

Operations on the histogram

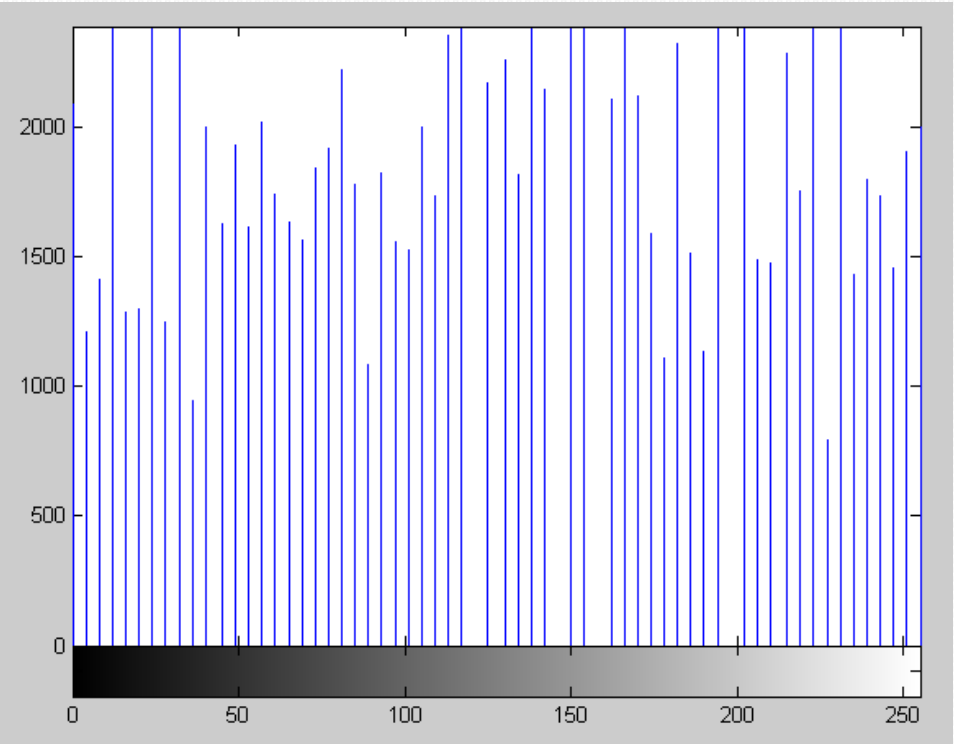
Image after histogram **stretching**

Image after histogram **equalizing**

Operations on the histogram



Histogram after **stretching**



Histogram after **equalizing**

Operations on the histogram

Image after histogram **stretching**

Image after histogram **equalizing**

Arithmetic operations

Arithmetic operations

Contextless operations – are performed on a single pixel, without considering neighbouring pixels

Context operations – are taking into consideration neighbouring pixels values

Arithmetic operations:

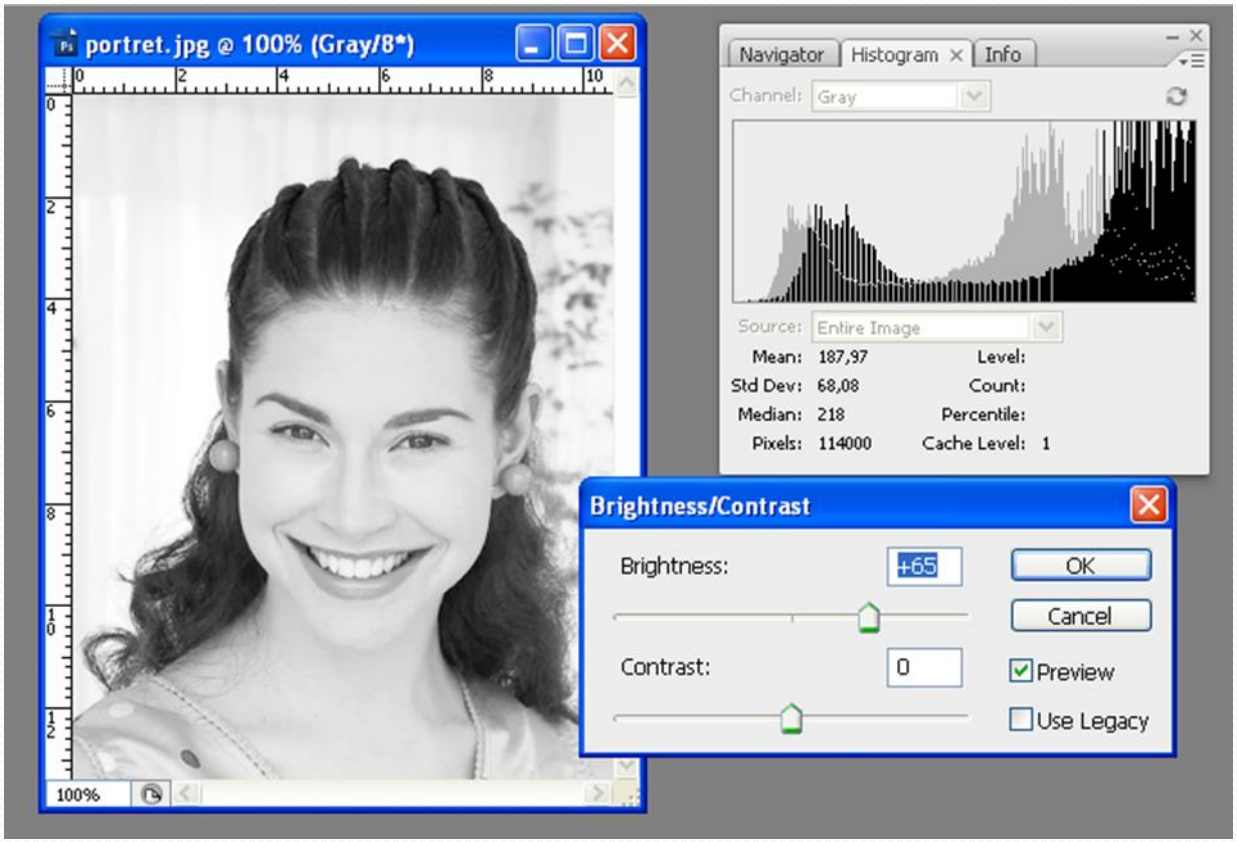
- adding a value,
- subtracting a value,
- logarithm,
- square root,
- power.

Arithmetic operations

Source image

Image after adding a value

Operations on the histogram



Adding a value

Brightening the image

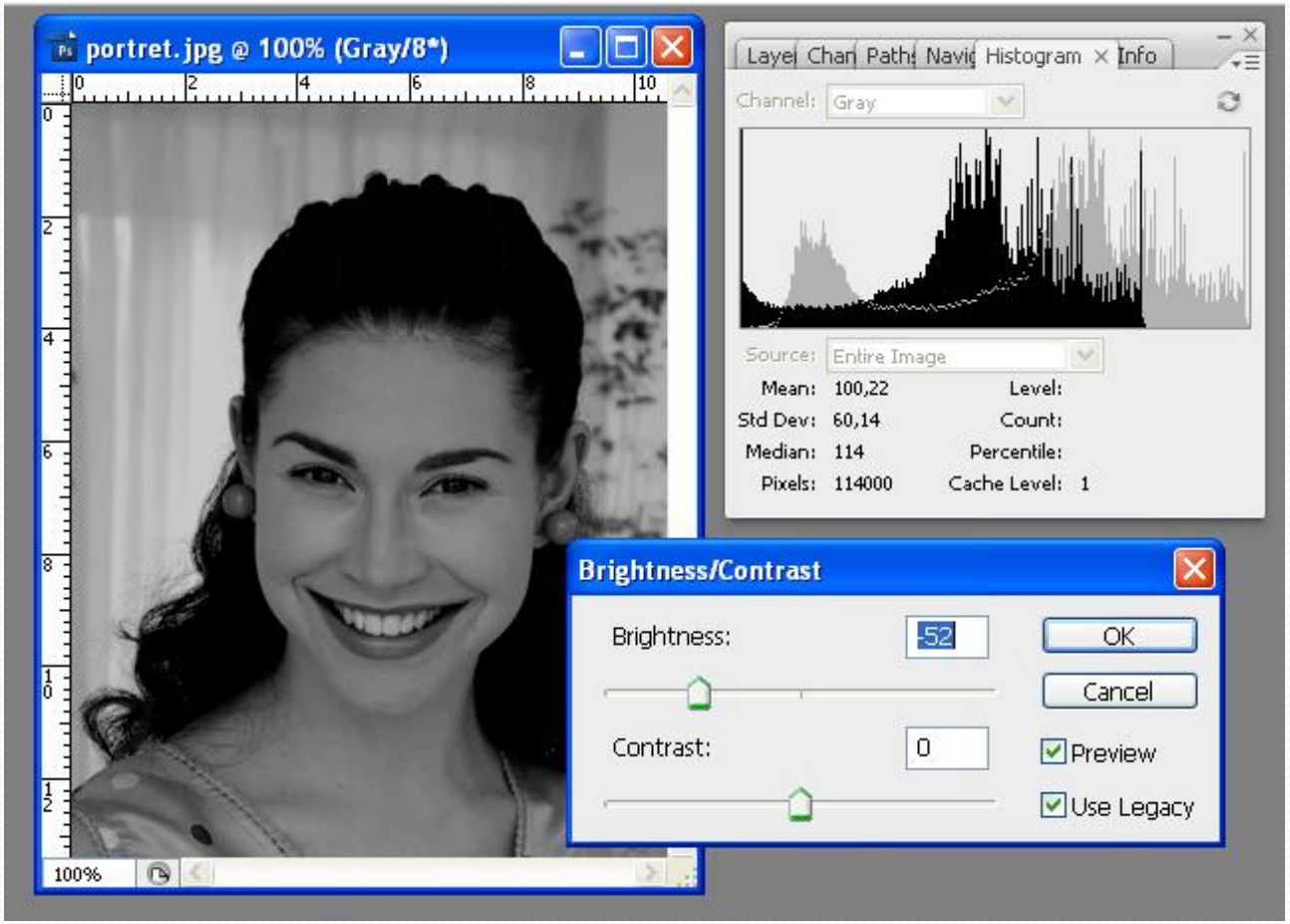
Shifting the histogram right

Arithmetic operations

Source image

Image after subtracting a value

Operations on the histogram



Subtracting a value

Darkening the image

Shifting the histogram left

Arithmetic operations

Source image

Image after multiplying

Arithmetic operations

Source image

Image after dividing

Arithmetic operations

The presented transformations of images are accompanied by the problem of possible exceeding the maximum or minimum value of the brightness.

If we receive a pixel with a value that exceeds the acceptable range as a result of a conversion, three approaches are possible:

- **saturation method** – values are trimmed at the acceptable extreme value;

  This is accompanied by a negative effect - the loss of some information about the image.

Arithmetic operations

- **modulo method** - treatment of points with a value greater than 255 as a result of modulo p (p = 256). Exceeding the maximum brightness while adding images may give a dark pixel as a result, and as a result of subtracting a pixel lighter than the darker one, we can receive a bright pixel.
- **normalization method** – dividing the resulting image by a maximum value obtained in the image after transformation.

Arithmetic operations

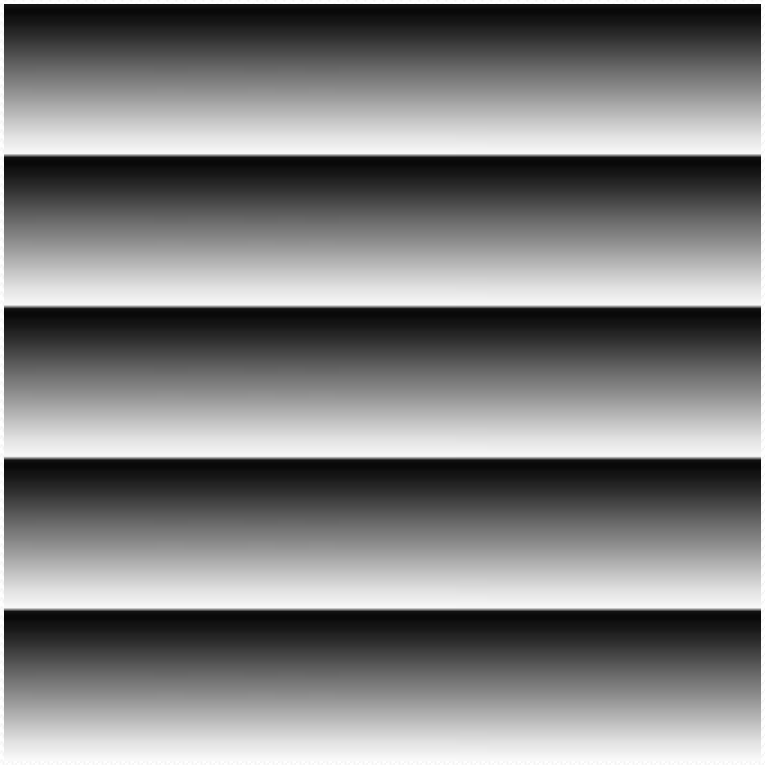
Saturation method

Source image

Image after multiplying

Arithmetic operations

Modulo method

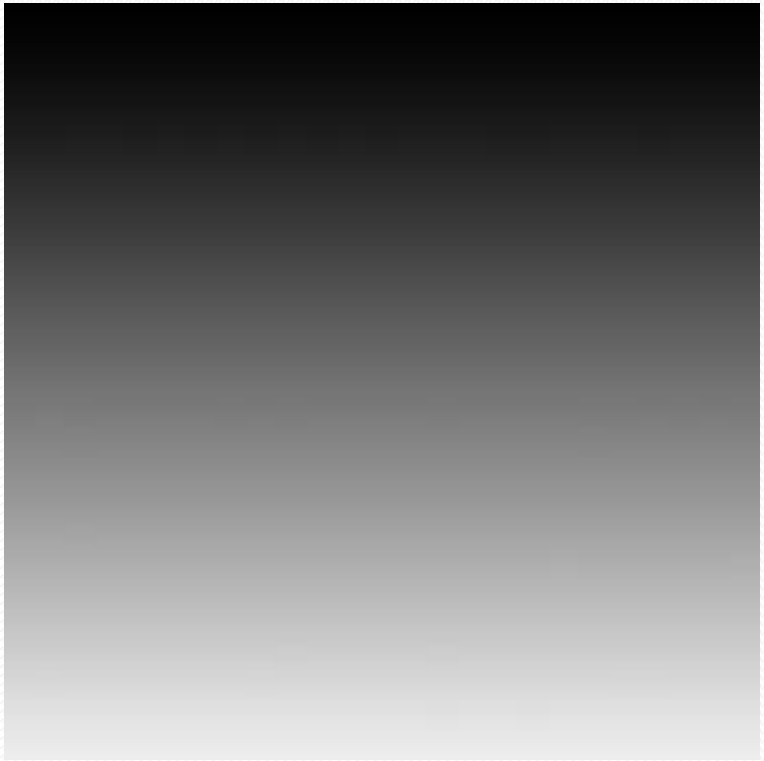


Source image

Image after multiplying

Arithmetic operations

Normalization method



Source image

Image after multiplying