Histograms and   
Image Statistics

Zbigniew Hulicki, Mikołaj Leszczuk, Jakub Nawała

# Abstract

This document is a exercise instruction, introducing students into basic histogram operations in image processing, using MATLAB.

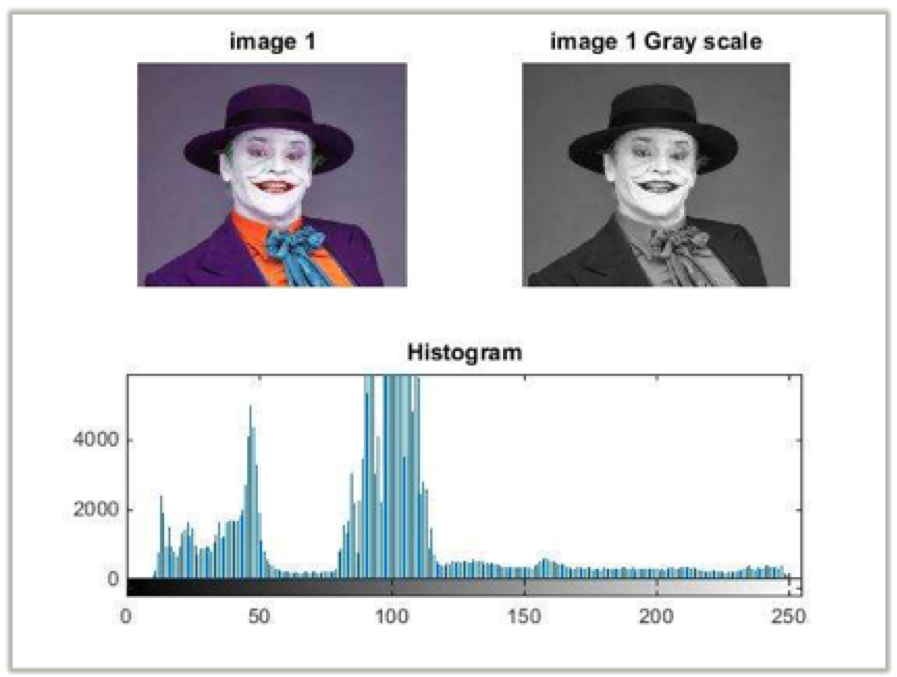
# What Is an Image Histogram?

An **image histogram** is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value.

It is important to note that Histogram IS NOT 1:1 property of the image. There are many images that have the same histogram.

## Applications of the Histogram

Image histograms are present on many modern [digital cameras](https://en.wikipedia.org/wiki/Digital_camera). Photographers can use them as an aid to show the distribution of tones captured, and whether image detail has been lost to blown-out highlights or blacked-out shadows.



## MATLAB Implementation Example

Please write a MATLAB script, which will:

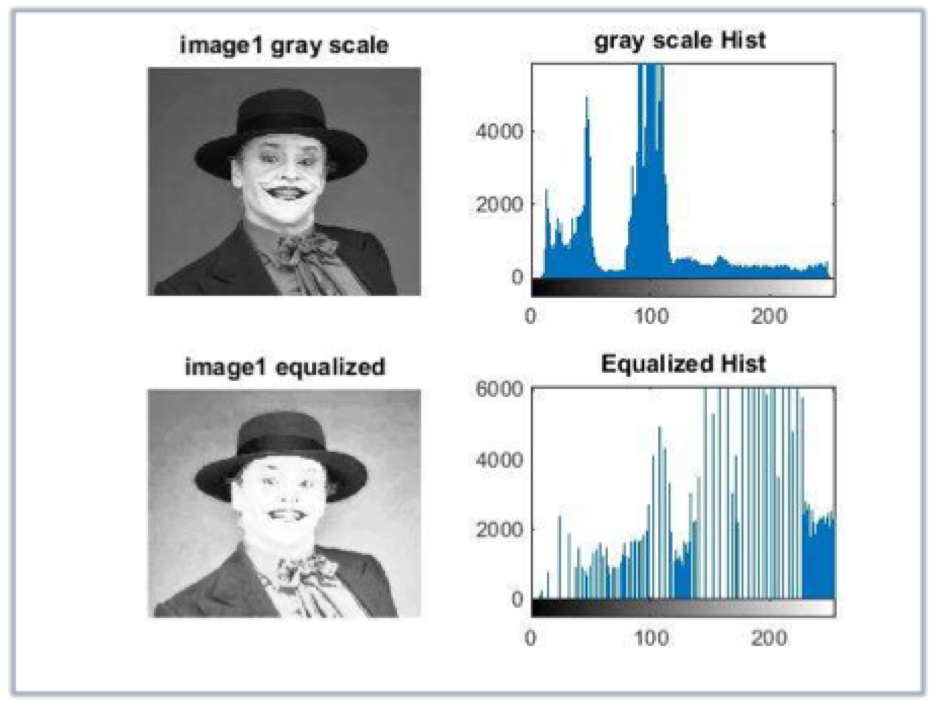
1. Read an original RGB image.
2. Convert the original RGB image to a grayscale.
3. Calculate a grayscale image histogram.
4. Display simultaneously:
   1. The original RGB image.
   2. The grayscale image.
   3. The grayscale image histogram plot.

# Histogram Matching (Specification)

There are many instances in which equalization produces quite undesirable effects. A closely related technique known as histogram matching (also known as histogram specification) is sometimes more appropriate and can be employed both as a means for:

* improving visual contrast,
* regularizing an image prior to subsequent processing or analysis.

The idea underpinning histogram matching is very simple. Given the original (input) image ***Iinput*** and its corresponding histogram ***px*(*x*)** we seek to effect a transformation ***f*(*x*)** on the input intensity values such that the transformed (output) image ***Ioutput*** has a desired (target) histogram ***pz*(*z*)**.



## MATLAB Implementation Example

Please write a MATLAB script, which will:

1. Read an original RGB image.
2. Convert the original RGB image to a grayscale.
3. Perform a histogram equalisation with a histogram length of 0 to 255.
4. Display simultaneously:
   1. The original RGB image.
   2. The grayscale image.
   3. The grayscale image histogram plot.
   4. The grayscale image after equalization of histogram.
   5. The equalised histogram.

# A Detector of Visually Dense Regions

Please imagine that you are implementing a detector of visually dense regions. Such a detector could be useful when performing image compression. It is crucial for a compression algorithm to know where are the regions containing a lot of information. Having this information, it can use more bandwidth to compress those (and leaving less bandwidth for other regions). Let us call the regions containing a lot of information *visually desne regions*.

Please implement a detector of visually dense regions. Do it by dividing an image into 32x32 pixels blocks and calculating a standard deviation for each block[[1]](#footnote-1). As an output, please return the position of the most visually dense region.

**Please report to the tutor if you have finished this exercise.**

# For Matlab Usage

1. **Create a new script in Matlab environment:**

Home >> New Script

1. **Save it as a .m file under the following path:**

for Windows: My documents >> MATLAB

for Linux distribution : home/MATLABWORKSPACE

Note that if you want to upload any file to the Matlab environment all of them should be placed in the same directory: MATLABWORKSPACE folder (e.g. ”pictures”).

1. **Now you can easily implement a code in the editor:**

Remember that before you start to run your program you should save all the changes you have done in the editor.

# Bibliography

1. “Digital Image Processing Using MATLAB”, Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins
2. “Fundamentals of Digital Image Processing: A Practical Approach with Examples in MATLAB”, Chris Solomon, Toby Breckon

1. If you work with a true colour image use the average standard deviation from all colour channels [↑](#footnote-ref-1)