Dpto. Ingeniería de Sistemas y Automática
Universidad de Málaga

COMPUTER VISION

EXERCISE 1.a: Introducing mVision

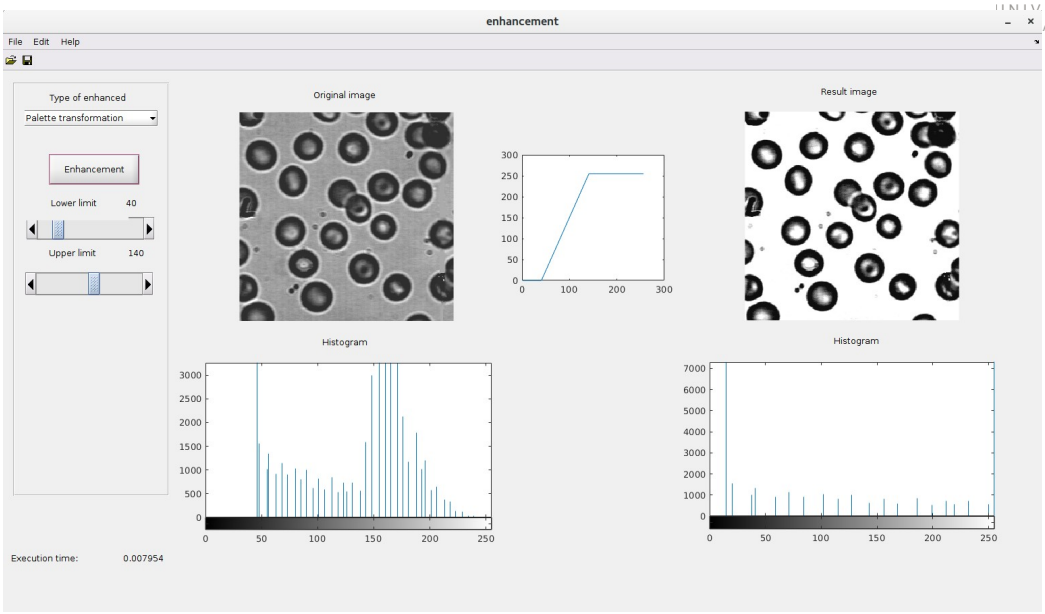
Concepts: Matlab GUIs

mVision consists of a set of Matlab GUIs (Graphical User Interfaces) aimed to show in action some of the typical algorithms introduced in Computer Vision courses. Please follow the next steps to configure it:

1. Go to: https://github.com/jotaraul/mVision. There are two possible ways to download it:
    a. Clicking on the "Clone or download" button, you can directly download it to your computer.
    b. If you are a *git* user, you can clone the repository into your computer with "git clone https://github.com/jotaraul/mVision". This enables additional options like proposing changes or easily uploading the code to the last version.
2. Add the directory "mVision" to your Matlab path.
3. Enjoy!

In this first exercise we are going to enhance and smooth images. Let's play a bit with mVision to gain insight into them, and then implement some code:

1. Launch the main mVision GUI by introducing *mVisionGUI* in the command window. Then launch the first GUI, *Smoothing*.
2. Load an image (for example, **kids**), and try the three available smoothing techniques: *Averaged environment*, *Gaussian filter* and *Median filter*.
    a. What is the effect of changing the mask (kernel) size?
    We get a blurrier image, but with less noise.
    b. Which technique is the slowest (largest execution time)? Why?
    The median filter, because it must first get the brightness of each neighbor pixel and then sort them in order to get the median.
3. Now launch the second GUI, *Enhancement*.
4. Load the images **kids**, **lily** and **blood**. Which one show the highest quality?
    According to the histogram, we can clearly see that lily has a higher contrast. However, it also seems to be a little bit pixelated.
5. Finally, play a bit with the two implemented enhancement techniques: *Palette transformation* and *Histogram equalization*, and discuss an example of image enhancement that you consider interesting.



The previous image shows a LUT function used to highlight the blood cells on top of the white background in order to perform a future blood cell counting.

Dpto. Ingeniería de Sistemas y Automática
Universidad de Málaga

EXERCISE 1.b: Image enhancement

Concepts: LUT, equalization and histogram specification

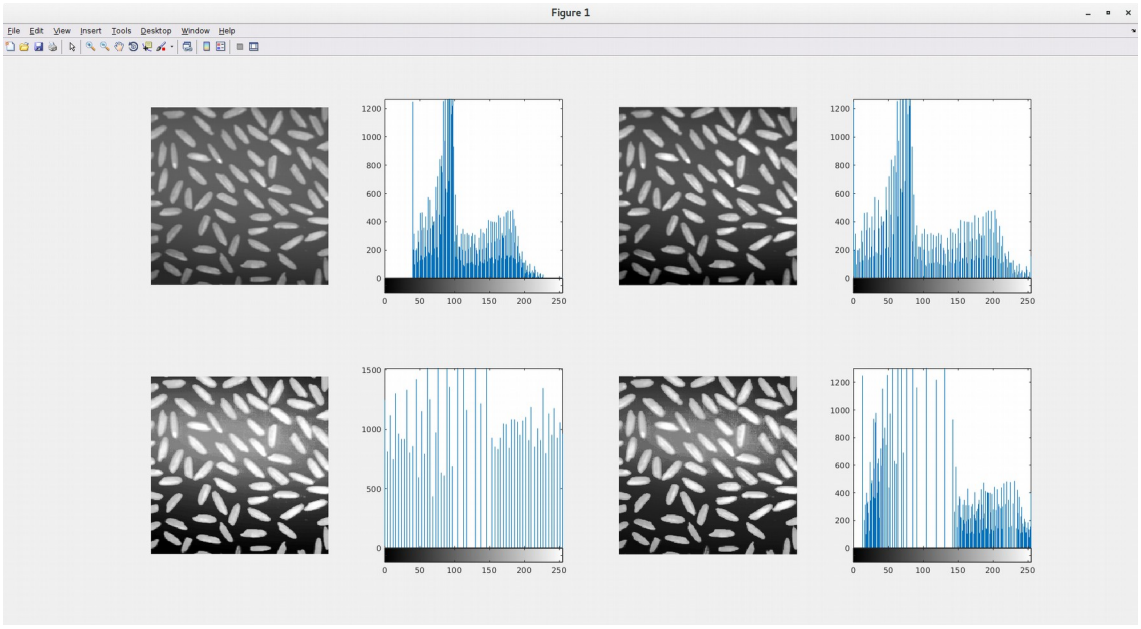
Implement a Matlab code showing a figure with 8 subplots (2x4), including the following:

1. Image **rice.tif** and its histogram.
2. Image **rice.tif** enhanced with a LUT (to design analyzing the initial histogram) and the resultant histogram.
3. Equalized image and its histogram.
4. Image with a specified histogram and the resultant histogram. Hint: The specified histogram can be retrieved from other image.

Useful functions

im = histeq (image, h_spec)	Modifies the **image** using the specified histogram in **h_spec**, or performs and equalization if it is omitted, returning the new image in **im**.
im = imadjust (image, [LOW_IN HIGH_IN], [LOW_OUT HIGH_OUT])	Transform the **image** using a lineal LUT with input intensity values **[LOW_IN HIGH_IN]** and output values **[LOW_OUT HIGH_OUT]**.

Results



EXERCISE 1.c: Image smoothing

Concepts: Noise in the image and filter types.

Implement a function that, given the name of an image, displays two figures (2x2) with the following:

1. Image with *Gaussian noise*, and its smoothed image employing a *Gaussian filter* with σ=0.5, a *Median filter*, and a *Neighborhood averaging filter* with a 3x3 kernel.
2. Repeat the previous images but with *salt & pepper* noise.

Analyze the obtained results.

Results (with **blood.tif**)

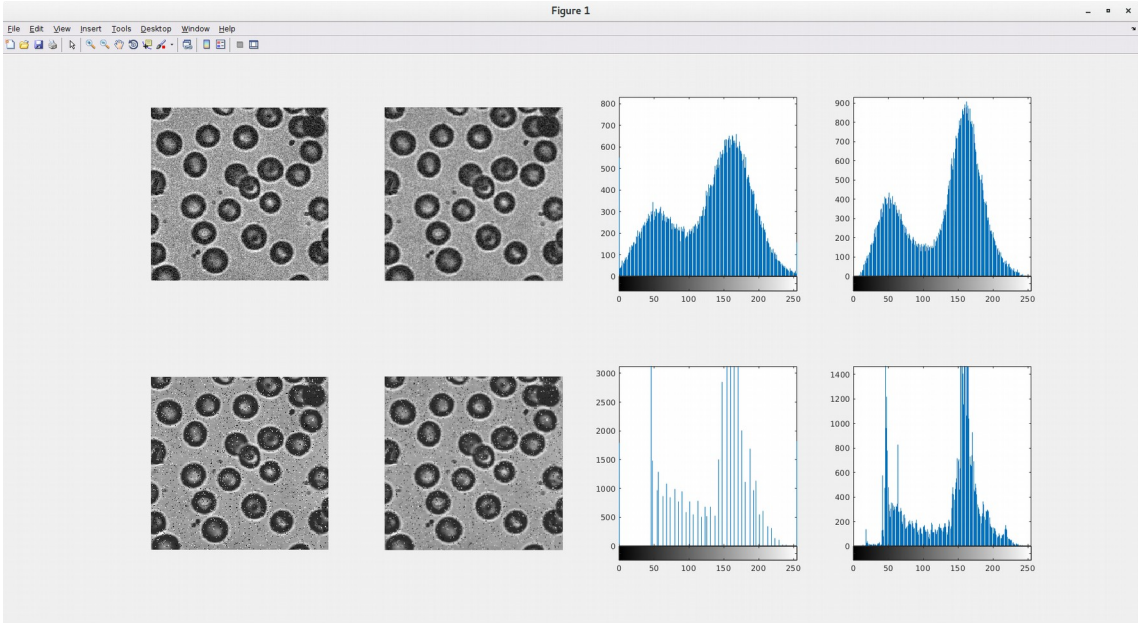


Illustration 1: Gaussian smoothing

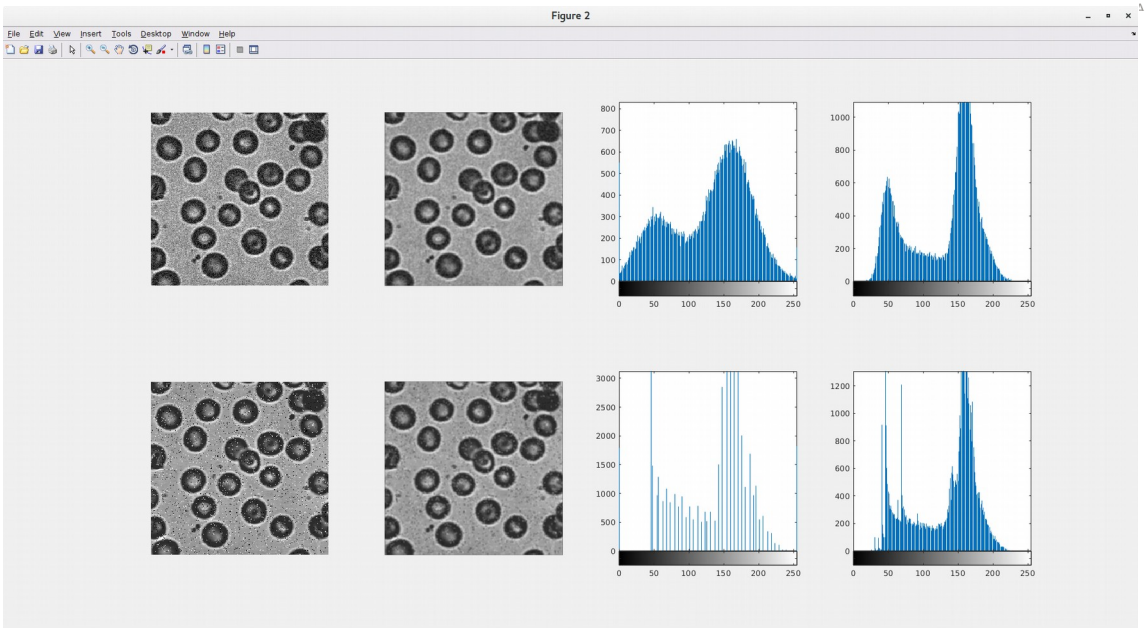


Illustration 2: Neighborhood averaging

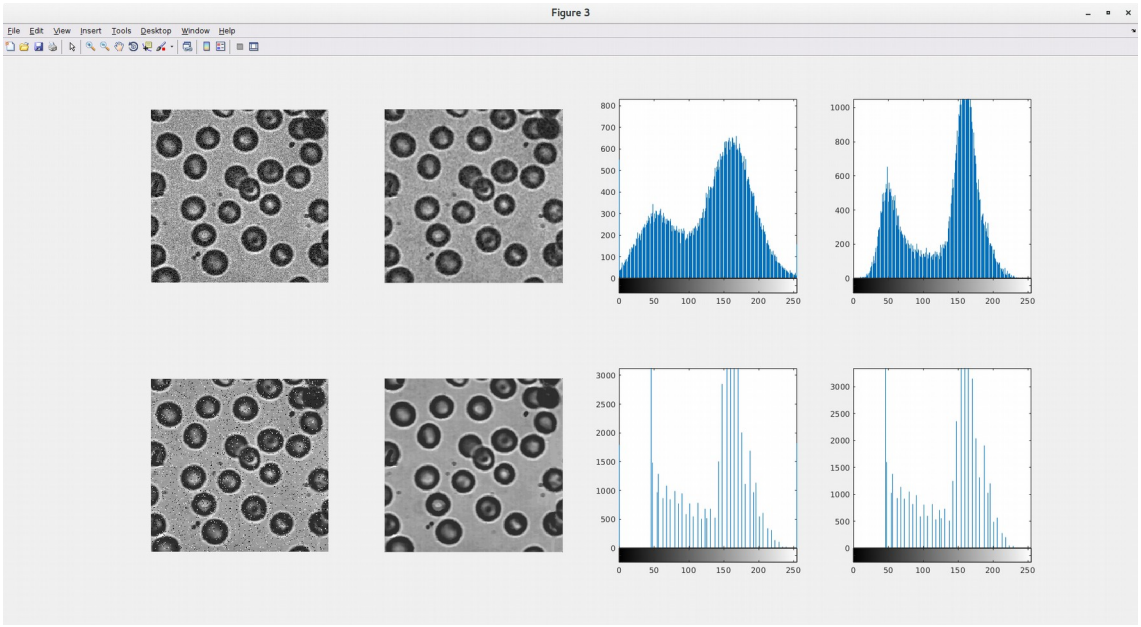


Illustration 3: Median filter