Filters and histogram equalization

The goal of this lab was to explore different image filters and histogram equalization techniques to process or enhance the quality of an image.

Task 1 was straightforward, and was solved simply by calling the `cvtColor` provided by OpenCV.

For task 2 I implemented **min** and **max** filters.
Some remarkable things that came out:

- kernel size of 5 is good with a **max** filter to remove electric cable in the background while keeping image quality high;
- **min** filter doesn't remove the cables, because they are darker than the background;
- increasing the kernel size, the image quality decreases;
- output image size is reduced, due to the convolution operation. An alternative (to keep the image size) was to apply 0-padding techniques;
- I discovered flags like `IMREAD_UNCHANGED` and `IMREAD_COLOR` that are used by the `imread` function to create the `Mat`. I left a comment in `task2.cpp` with more details.

In Task 3, I extended Task 2 to include **median** and **gaussian** smoothing filters provided by OpenCV. The gaussian smoothing appears softer and more "natural", while the median gives "harder" outputs.

For Task 4 I started implementing the `calcHistImg` function and was happy to see it work properly with 256 bins (same result as `cv::calcHist` + visualization). However, I didn't expand the function to support other numbers of bins and ultimately used the OpenCV function to solve the task. The histogram shows strong shadows and no highlights in the image. This justifies the equalization performed in Task 5.

In Task 5, the input image was histogram-equalized, and the result shows a much better dynamic range. The histogram is flattened and distributed along all the bins.

Alberto Zerbinati, 26/03/2023