

Computer vision 2024 – Lab 2

Task 1

In task 1 I used the `cv::cvtColor()` function to with the `cv::COLOR_BGR2GRAY` code to convert the image to grayscale. I also included a couple of lines of code so that this procedure is generalized for any image. This way i could use this to covert other images in the later tasks. Attached is the resulted image.



Figure 1: Garden grayscale.

Task 2

For the max filter I implemented it using a quadruple for loop. The first two loops are used to access the pixels of the image matrix. The second two loops are used to loop over the $N \times N$ kernel. During the second two loops, the current detected max value inside the kernel is stored. After the whole kernel is looped over, the current pixel is set to the stored max value. The minimum filter is implemented the same way. Except this time, the minimum



Figure 2: Astronaut min filter.



Figure 3: Astronaut max filter.

value is stored and selected. Attached are the best results obtained using the maximum and minimum filters.



Figure 4: Garden max filter.



Figure 5: Lena max filter.

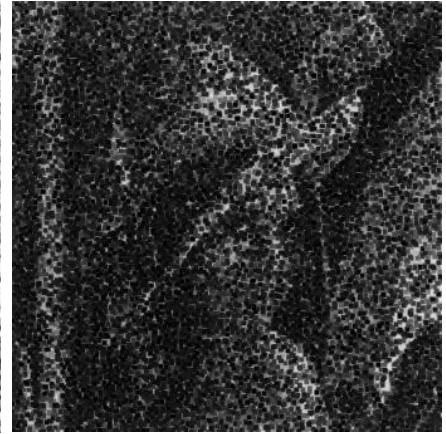


Figure 6: Lena min filter.

The results are not very impressive, however this is not surprising due to the amount of corruption in the images (as talked about in class). Because of the low resolution of the astronaut image even with kernel 3, “big” white/black boxes appear.

For the garden image, we obviously only need to look at the max filter when wanting to remove the cables.

Task 3

The median filter was more or less implemented in the same way as the max and min filter. Only this time all the values in the kernel was stored in a vector that finally was sorted and the middle value was selected.

The results were, even though not very impressive, a lot better than the results from the max/min filters in removing the salt and pepper noise. This was as expected from theory. The electrical cables were obviously more visible than in the max filter.



Figure 7: Garden median.



Figure 8: Astronaut median.



Figure 9: Lena median.

Task 4

The gaussian filter was implemented using the `cv::GaussianBlur()` function. As expected from theory, the filter does not work as well as the median filter on the salt and pepper noise. Also, the cables are more visible than for the max filter.



Figure 10: Garden gaussian.



Figure 11: Lena gaussian.



Figure 12: Astronaut gaussian.

Task 5

The histogram for the “Garden_grayscale.jpg” image was calculated using the `cv::calcHist()` function. Also the `cv::minMaxLoc()` function was used to scale the output. The `cv::rectangle()` function was then used to draw the histogram. Attached is the result using 256 bins. The result clearly shows an overwhelming amount of darker grayscale values in the image.

Task 6

The “Garden_grayscale.jpg” image was equalized using the `cv::equalizeHist()`. Due to the amount of dark grayscale values the image is clearly brightened up by the equalization. And the result is a much clearer and balanced picture.

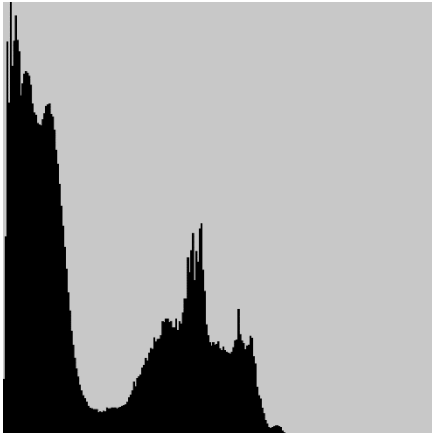


Figure 133: Histogram of original Garden_grayscale.



Figure 14: Equalized image.