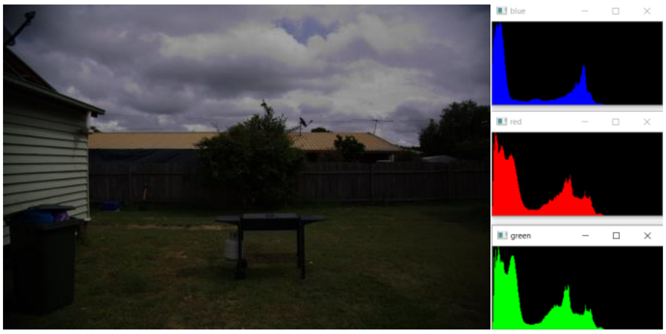
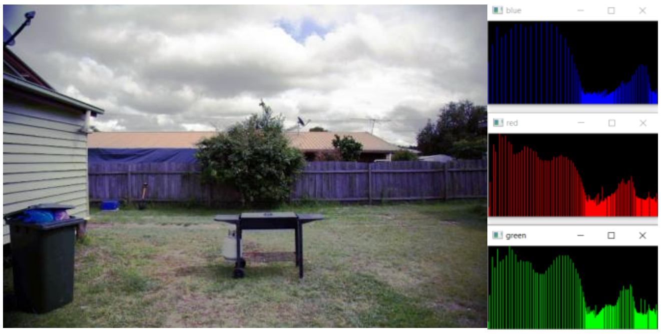
Computer vision 2022: Nadezhda Manuilova ID:2049581

1) Histogram computing and equalization

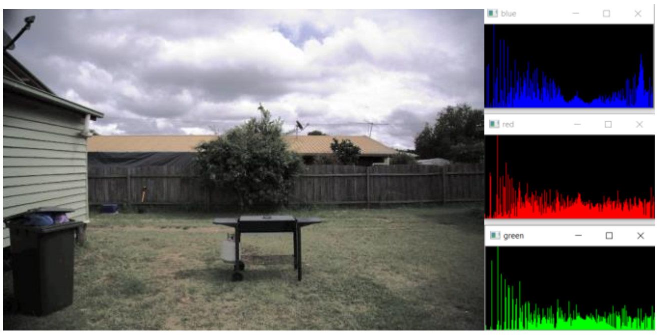
Original the image barbecue.png is dark. We compute the histogram of the image using the OpenCV function cv::calcHist()for the three different color components (B,G,R). Showimage() function show the result.



For better distribution of the pixel we use function cv::equalizeHist(). We can see that the image is brighter but not good.

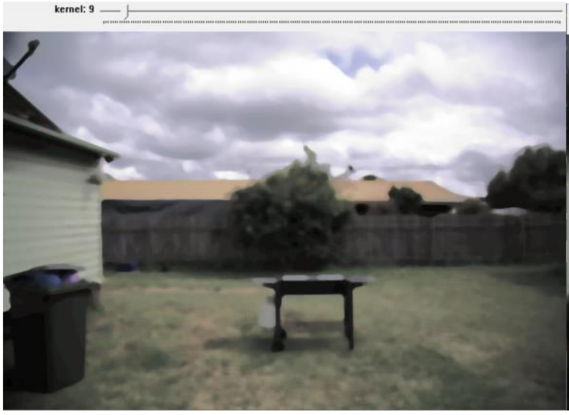


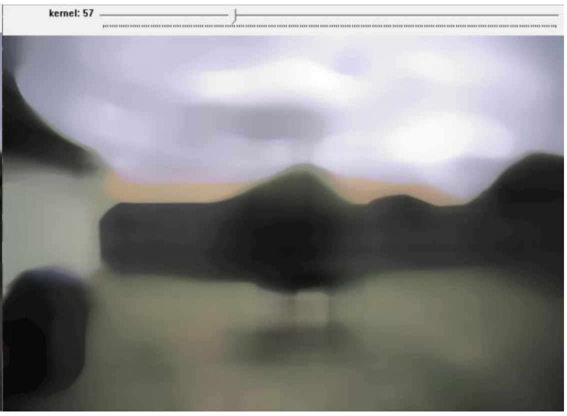
Next we convert the image in the CIELab color space. We have one channel for the luminance and only two channel. We again first compute the histogram and then we equalize only the luminance channel. Now the picture much more better.



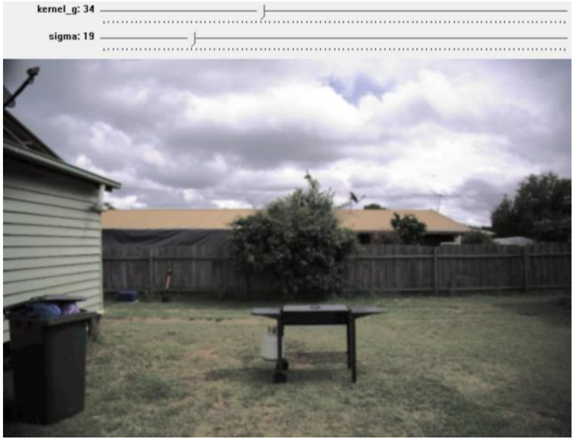
2) Image filtering

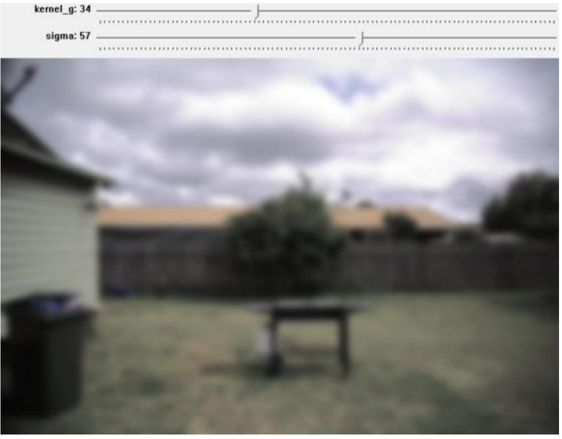
Median filter(non linear filter): To implement it we use the function cv::medianFilter() and the parameter we can vary is the dimension of the kernel. The higher is the kernel size the higher is the number of details which we can recognize.


kernel: 9

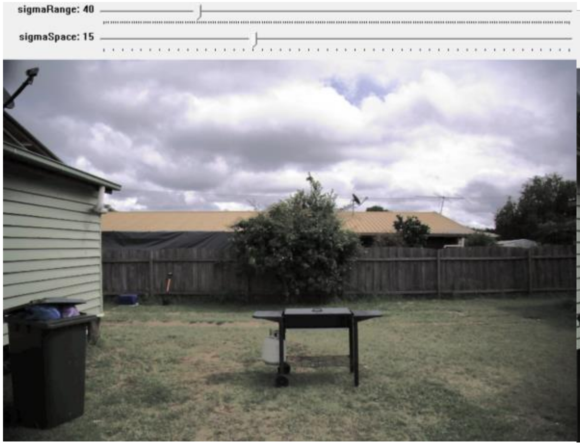

kernel: 57

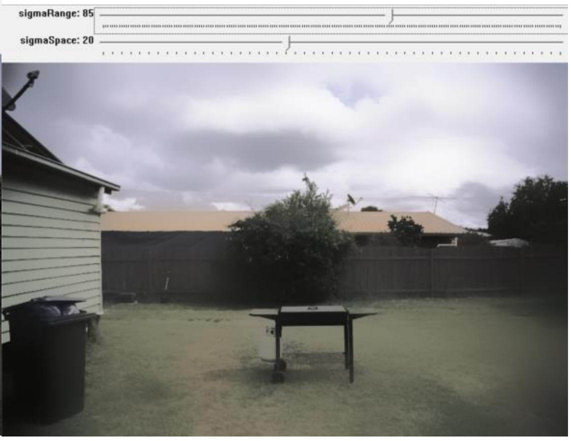
Gaussian Filter(linear filter): We use the function cv::GaussianBlur() to reduce the noise in an image. We can see that a small value of sigma allow to preserve the most quantity of the detail but a limited smoothing but a large value of sigma allows a strong smoothing and the most details are destroyed.


kernel_g: 34
sigma: 19


kernel_g: 34
sigma: 57

Bilateral Filter(non linear filter): We use the function cv::bilateralfilter(). In the left image with a smaller value of sigma range, details are well preserved but in the right image we have that the details looks blurry. The image have soft effect.


sigmaRange: 40
sigmaSpace: 15


sigmaRange: 85
sigmaSpace: 20