EE417

POST-LAB #1 REPORT

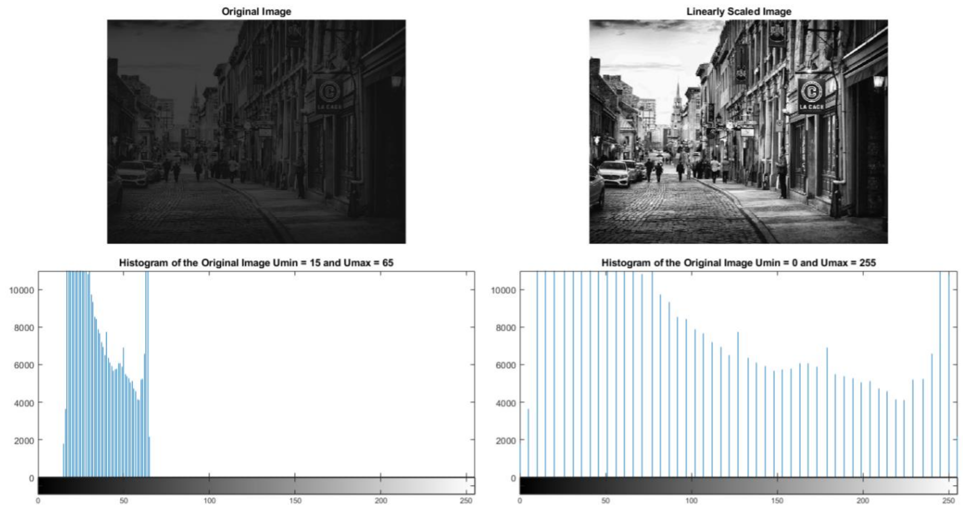
1. Linear Scaling

In this section we used a gradation function as follows:

$$g(u) = b(u + a) \text{ where } a = -umin,\ b = Gmax/(umax-umin) \text{ and } Gmax = 255$$

lab1locbox function takes an image and it checks its channel, if it is has 3 channels it converts the image into grey scale. In order to make computations, it is converted into double. After the scaling we turned into uint8 in order to display the image.

```
function [J] = lab1linscale(img)
    Dimg = double(img);
    [row,col,ch] = size(img);
    if(ch==3)
        doubleImg = rgb2gray(img);
    end
    umin = min(Dimg(:));
    umax = max(Dimg(:));
    Gmax = 255;
    a = -umin;
    b = Gmax/(umax-umin);
    K = b.*(Dimg+a);
    J = uint8(K);
end
```

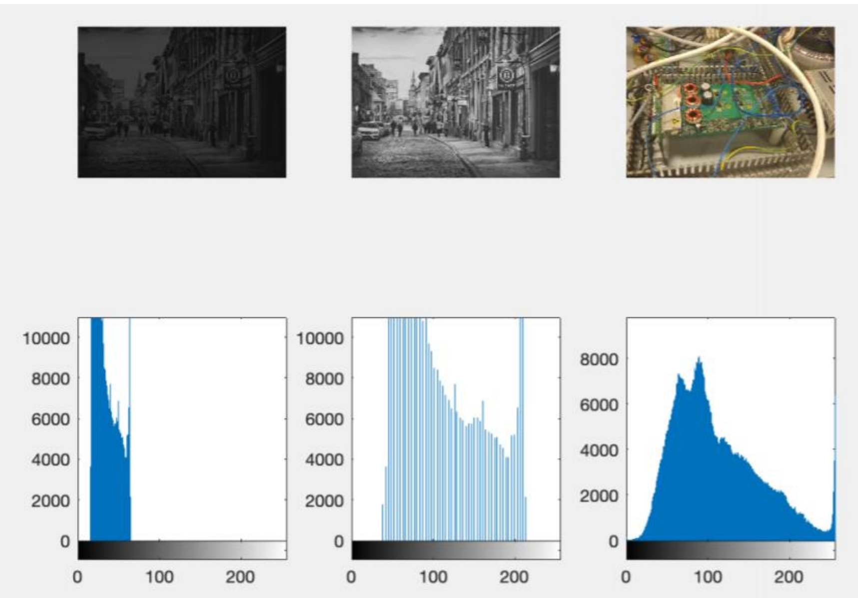


If we look at the histogram of the image we can say that it actually seems scaled like accordion over the whole region. In the image it has darker pixel values, after the scaling it introduces the lighter pixels which results in a brighter image.

2. Conditional Scaling

In this section we firstly computed mean and standard deviation of the images and then by conditional scaling we obtain an image that has the same mean and variance values by the following gradation function: $g(u) = b(u + a)$ (2) where, $a = \mu I * \sigma J / \sigma I - \mu J$ and $b = \sigma I / \sigma J$

```
function [J] = lab1condscale(img1,img2)

    [row1,col1,ch1] = size(img1);
    [row2,col2,ch2] = size(img2);

    if(ch1==3)
        img1 = rgb2gray(img1);
    end

    if(ch2==3)
        img2 = rgb2gray(img2);
    end

    Dimg1 = double(img1);
    Dimg2 = double(img2);

    mean1 = mean(Dimg1(:));
    mean2 = mean(Dimg2(:));

    std1 = std(Dimg1(:));
    std2 = std(Dimg2(:));

    a = mean1*(std2/std1)-mean2;
    b = std1/std2;
    K = b.*(Dimg2+a);
    J = uint8(K);
end
```



If we compare the histograms of the images, we observe that current image has darker pixel values by conditional scaling it converted a brighter image because it has higher mean pixels that transforms mean of the image into a higher mean values that results a brighter image.

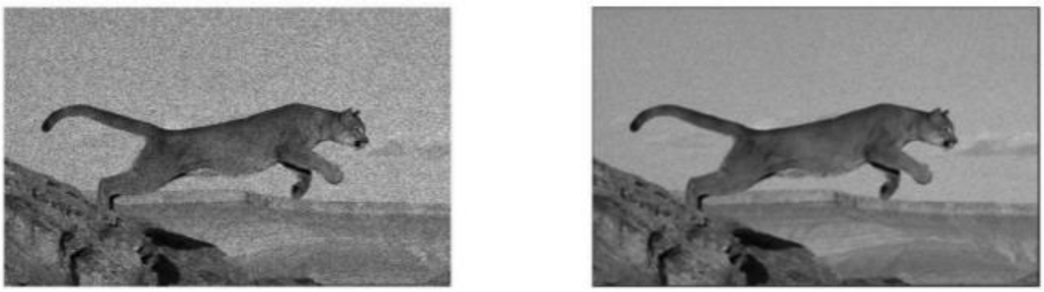
3. Box Filter

In this section we applied KxK window in order to reduce noise in the image by convolving it with a sliding window WP of size (2k+1)×(2k+1). Box filter is realized by replacing each pixel of an image with the average of its neighborhood as follows:

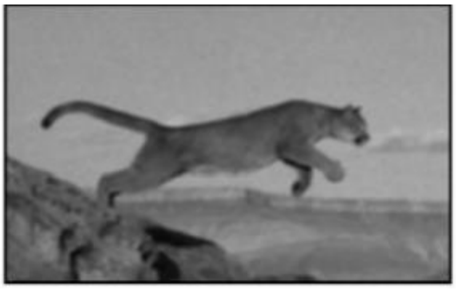
$$\mu_{W_p(I)} = \frac{1}{(2k+1)^2} \sum_{i=-k}^{+k} \sum_{j=-k}^{+k} I(x+i, y+j)$$

$$J(p) = \mu_{W_p(I)}$$

```
function [J] = lab1locbox(img)

    [row,col,ch] = size(img);
    A = zeros(size(img));
    k = 5;

    if(ch==3)
        img = rgb2gray(img);
    end

    Dimg = double(img);

    for i = k+1:1:row-k-1
        for j = k+1:1:col-k-1
            subIm = Dimg(i-k:i+k,j-k:j+k);
            value = mean(subIm(:));
            A(i,j) = value;
        end
    end

    J = uint8(A);

end
```



Applying Box Filter with K = 3

Applying Box Filter with K = 5

As we it can be seen above, increasing K is reducing the noise in the image but it also makes the image more blurry because box filter averages the pixels and assign this value. When we incerase window size it averages a bigger region that causes blur.

4. Local Minimum/Maximum Filter

We applied minimum/maximum filter that replaces the pixels with local minimum/maximum of its neighbours.

$$J(p)_{max} = max\{I(x+i, y+j) : -k \leq i \leq k \ \wedge \ -k \leq j \leq k\}$$

$$J(p)_{min} = min\{I(x+i, y+j) : -k \leq i \leq k \ \wedge \ -k \leq j \leq k\}$$

```
function [Jmax,Jmin] = lab1locmaxmin(img)

    [row,col,ch] = size(img);
    A = zeros(size(img));
    B = zeros(size(img));
    k = 5;

    if(ch==3)
        img = rgb2gray(img);
    end

    Dimg = double(img);

    for i = k+1:1:row-k-1
        for j = k+1:1:col-k-1
            subIm = Dimg(i-k:i+k,j-k:j+k);
            value1 = max(subIm(:));
            value2 = min(subIm(:));
            A(i,j) = value1;
            B(i,j) = value2;
        end
    end
    Jmax = uint8(A);
    Jmin = uint8(B);
end
```

Applying Local Minimum/Maximum Filter with K = 3

Applying Local Minimum/Maximum Filter with K = 10

Applying Local Minimum/Maximum Filter with K = 30

By applying maximum filter the image gets brighter, on the other hand by applying minimum filter the image gets darker. But when K increases it introduces blur and noise in the image because it assigns a same max or min value for the whole region.